POLICE CALLS FOR SERVICE EXPLORATORY DATA ANALYSIS

Data Science Track: Capstone Project -1

ABSTRACT

The 911 Public Safety Answering Point (PSAP) service area boundaries in California, according to the National Emergency Number Association (NENA), is a facility equipped and staffed to receive 9-1-1 calls. The service area is the geographic area within which a 911 call placed using a landline and is answered at the associated PSAP.

New functionalities on other systems developers now include calls from mobile phone and cell texting. This project will only address calls from landlines.

Author: Franklin Cid

Mentor: Milad Toutounchian

SUMMARY

Objective

Calls for service to law enforcement agencies generally includes calls to “911” for emergency assistance and occasionally includes calls to non-emergency numbers. These data are typically maintained in law enforcement computer-aided dispatch systems (CAD). The objective of this study is to analyze the dataset in order to identify the geographical locations (districts), times of the year, priority calls, and call types (reasons for the emergency call) of the most priority calls.

Goals

The goal in this project is to analyze emergency calls in order to give PSAPs management insights into their call volume. With this insight, their scheduling process can be optimized to address call volumes, anticipate call types, and expect the origin of the call.

This task is an exploratory data analysis project. As such, the author wants to find the answers to the problems below:

1) What district has the greatest <High> priority calls for (Jan/2015, Jan/2016, Jan/2017)?
2) What are the top 4 priority calls for 3 years (2015-2017)?
3) What month(s) did the most police calls occur for each year (2015, 2016, 2017) based on each priority call (High, Medium, Low, and Non-Emergency)?
4) What are the top 3 callTypes (type of emergency calls) for (Jan/2015, Jan/2016, Jan/2017)?
5) What is the percentage of emergency calls and non-emergency calls?
6) What are the percentages of High, Medium, and Low priorities from the emergency calls?

Prospective Clients

Prospective clients include systems developers for emergency calls tracking system (e.g. West/ECaTS), PSAPs (the facilities, usually on the county level, that actually receive and answer the emergency and non-emergency calls all over the U.S.), and local/city government that require deeper analysis and insights of these calls.

Data: Acquisition

The dataset that will be used for this project comes from the City of Baltimore in Maryland. This contains 3.86 million rows with eight columns/row. Each row comprises a single call. The data is updated regularly and can be downloaded from the city website: data.BaltimoreCity.gov.

Data: Preliminary Assessment

The Baltimore, Maryland 911 Police Calls for Service appears to be a raw data from actual calls. A close look shows the observations:

a. There are 8 columns but only 4 will be used for this study. The columns “recordId” and “callNumber” appear to be system-generated and will not have any use for this study. The columns “incidentLocation” and “location” refer to the address of the call but are not also needed in this study. Latitude and longitude are also included in location.

```
c911.head(2)
```

	recordId	callDateTime	priority	district	description	callNumber	incidentLocation	location
0	2749202	08/10/2017 03:25:00 PM	Low	ND	LARCENY	P172221666	3100 ST PAUL ST	3100 ST PAUL ST\nBALTIMORE, MD\n(39.326142, -7...
1	2785533	08/23/2017 12:53:00 PM	Medium	NE	911/NO VOICE	P172351302	3300 BELAIR RD	3300 BELAIR RD\nBALTIMORE, MD\n(39.321622, -76...

b. The column “description” appears to be the reason for the call or type of call, herein referred to as callType. For clarity, this column name will be replaced with “callType”.

c. The columns needed for this study are: “callDateTime”, priority”, “district”, and “callType”.

d. There are three categorical data namely: priority, district, and callType. The callDateTime will be used to do the data groupings, index, etc.

e. Based from the three categories above,

   1) district – the data used is the same month (January) for 3 consecutive years (2015, 2016, and 2017).

   2) priority – the data used is 3 full consecutive years (2015, 2016, and 2017).

   3) callType – the data used is the same month (January) for 3 consecutive years (2015, 2016, and 2017).

f. Looking at the raw data, the author observed that there were data inconsistencies identified below. Hence, data wrangling is required. Data wrangling procedure is defined in a separate Python notebook, “Data/ 911PoliceCalls4Svc.EDA.wrangle.ipynb”.

   4) Empty/NaN/Null values

   5) Mix of Uppercase/Lowercase

   6) Data error - typographic, extra space(s)

g. The dataset provides a priority value of “Non-Emergency”. There is also an “Emergency” column value totaling only 1120. However, the column values of “High”, “Medium”, and “Low” are there as expected. Since the emergency calls is insignificant (1120), the author then decided to categorize them as high priority calls.

   In a mathematical formula, Emergency = High + Medium + Low.

h. A fifth priority observed is 'Out of Service' with an occurrence of 880. Since this number is negligible and may indicate a problem with the call, the author decided to drop that priority.

Project Outline:

The project was performed in 3 separate steps:

1. Dataset was downloaded from the data source, formatted properly, and filtered accordingly.
2. Dataset was wrangled, cleaned up, then prepared for creating charts and visualization.
   a. Filtered data: 1) [01/2015 – 12/2017], [01/2015, 01/2016, 01/2017]
   b. Empty/NaN/NULL values – checked and fixed
   c. Uppercase/Lowercase – made all callTypes Upper()
   d. Typographical error, extra space(s) – checked and fixed
   e. Worked only on top 5 district

f. Worked only on top 4 priority

g. Worked only on top 5 callType

3. Created charts and graphs. With dataframe, matplotlib, numpy, and Seaborn packages, used appropriate stat/plot functions to answer the problems stated in the Goals section.

Solution / Approach:

The author analyzed the dataset using Python and visualization libraries and numerical packages including:

1. numpy
2. matplotlib plot
3. pandas dataframe
4. seaborn

The Python 3.6.4 environment was used to create all the codes for this project.

#0 Import packages and visualization libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
% matplotlib inline
```

#1 read 911 calls

```
c911 = pd.read_csv('Data/w911PoliceCalls_BaltimoreMD.csv')
```

```
#1a Check the data structure
c911.head(2)
```

	recordId	callDateTime	priority	district	description	callNumber	incidentLocation	location
0	2749202	08/10/2017 03:25:00 PM	Low	ND	LARCENY	P172221666	3100 ST PAUL ST	3100 ST PAUL ST\nBALTIMORE, MD\n(39.326142, -7...
1	2785533	08/23/2017 12:53:00 PM	Medium	NE	911/NO VOICE	P172351302	3300 BELAIR RD	3300 BELAIR RD\nBALTIMORE, MD\n(39.321622, -76...

```
c911.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3858058 entries, 0 to 3858057
Data columns (total 8 columns):
recordId            int64
callDateTime        object
priority            object
district            object
description         object
callNumber          object
incidentLocation    object
location            object
dtypes: int64(1), object(7)
memory usage: 235.5+ MB
```

#2 Pull c911 into a df, just including needed columns, and set the index. Change col name 'description' to 'callType' (reason for the call).

```
df911 = pd.DataFrame(c911, columns=['callDateTime', 'priority', 'district', 'description'])
df911.columns = df911.columns.str.replace('description','callType')
df911.head(6) #chk
```

	callDateTime	priority	district	callType
0	08/10/2017 03:25:00 PM	Low	ND	LARCENY
1	08/23/2017 12:53:00 PM	Medium	NE	911/NO VOICE
2	08/07/2017 06:57:00 PM	Medium	ED	911/NO VOICE
3	08/18/2017 09:05:00 PM	Low	NW	AUTO ACCIDENT
4	06/27/2015 03:12:00 PM	Non-Emergency	ND	LOUD MUSIC
5	08/23/2017 10:09:00 AM	High	NW	AUDIBLE ALARM

#3 Cast callDateTime => dateTime (no date fx available), new col=dT

df911['dT'] =pd.to_datetime(df911.callDateTime)

##4 Set the INDEX and SORT before filtering the target datasets.

```
df911.set_index('dT', drop=True)
df911.sort_values(by='dT', inplace=True)
df911.head(6) #chk
```

	callDateTime	priority	district	callType	dT
1106565	01/01/2015 12:01:00 AM	High	NW	Dischrg Firearm	2015-01-01 00:01:00
357881	01/01/2015 12:01:00 AM	High	SW	Shooting	2015-01-01 00:01:00
359329	01/01/2015 12:01:00 AM	Low	SW	Other	2015-01-01 00:01:00
359125	01/01/2015 12:02:00 AM	Low	SW	Subject stop	2015-01-01 00:02:00
359340	01/01/2015 12:02:00 AM	High	SW	Dischrg Firearm	2015-01-01 00:02:00
358628	01/01/2015 12:02:00 AM	High	NE	Dischrg Firearm	2015-01-01 00:02:00

#5 Create needed columns for manipulating/grouping data later.
#Calculate the dateparts for yr, mo, and dy.
#Now, Drop callDateTime col.

```
df911['yr'] = ''; df911['mo'] = ''; df911['dy'] = ''
df911.yr = pd.DatetimeIndex(df911.dT).year
df911.mo = pd.DatetimeIndex(df911.dT).month
df911.dy = pd.DatetimeIndex(df911.dT).day
```

```
#Drop callDateTime col now.
df911 = df911.drop('callDateTime', axis=1) #3858058 rows × 7 columns
df911.tail() #chk
```

	priority	district	callType	dT	yr	mo	dy
2492563	Low	NE	ESCORT	2018-10-10 11:43:00	2018	10	10
2492696	Medium	ED	DISORDERLY	2018-10-10 11:44:00	2018	10	10
2492673	Low	NW	CHECKWELLBEING	2018-10-10 11:44:00	2018	10	10
2492546	Medium	NE	FAMILY DISTURB	2018-10-10 11:45:00	2018	10	10
2492589	High	NE	SILENT ALARM	2018-10-10 11:45:00	2018	10	10

Data Wrangling

```
#6 #Data wrangle: df911
##6a Filter: 01/2015 - 12/2017 (3 full years)
##6b Empty/NaN/NULL
##6c Uppercase/Lowercase - make all callTypes Upper()
##6d Mistyped - extra space,
##6e Work only on top 5 district - dfDistrict (Jan/2015, Jan/2016, Jan/2017)
##6f Work only on top 4 priority - dfPriority (3 full years of 2015, 2016, 2017)
##6g Work only on top 5 callType - dfCallType (Jan/2015, Jan/2016, Jan/2017)
```

```
##6a Filter: 01/2015 - 12/2017 (3 full years)
df911 = df911.loc[(df911.dT >= '2015-01') & (df911.dT < '2018-01')]  ##zap!
df911.tail()
```

	priority	district	callType	dT	yr	mo	dy
3083878	Low	CD	OTHER	2017-12-31 23:58:00	2017	12	31
873473	Non-Emergency	NE	Business Check	2017-12-31 23:58:00	2017	12	31
140102	Medium	NE	FAMILY DISTURB	2017-12-31 23:58:00	2017	12	31
2161528	Medium	ND	DISORDERLY	2017-12-31 23:59:00	2017	12	31
3125919	Medium	SE	911/NO VOICE	2017-12-31 23:59:00	2017	12	31

```
##6b Empty/NaN/NULL   #priority missing(NULL)
dfPnull = df911[df911.priority.isnull()]
dfPnull.head()
```

	priority	district	callType	dT	yr	mo	dy
3253354	NaN	SW	BURGLARY	2015-04-27 11:13:00	2015	4	27
3362416	NaN	SE	COMMON ASSAULT	2015-04-27 11:14:00	2015	4	27
3249583	NaN	NW	ROBBERY ARMED	2015-04-27 11:23:00	2015	4	27
23652	NaN	NW	PURSE SNATCH	2015-04-27 11:24:00	2015	4	27
1106645	NaN	NW	PURSE SNATCH	2015-04-27 11:24:00	2015	4	27

```
##6b Empty/NaN/NULL
#Get unique list of callTypes with counts. Focus on Top 10
dfPnull.callType.value_counts().head(10)
```

```
COMMON ASSAULT     1883
BURGLARY           1106
DISORDERLY          708
ARMED PERSON        338
AGGRAV ASSAULT      296
DESTRUCT PROPTY     253
LARCENCY            253
DISCHRG FIREARM     216
FAMILY DISTURB      203
HIT AND RUN         104
Name: callType, dtype: int64
```

```
##6b Empty/NaN/NULL #Determine what priority is likely for this callType >>> Medium
dfPnull10 = df911[df911.callType =='COMMON ASSAULT']
pd.crosstab (dfPnull10.callType, dfPnull10.priority)
```

priority	High	Low	Medium	Out of Service
callType				
COMMON ASSAULT	322	8	132771	33

#Used a similar code for the rest of the Top 10 callTypes to find the likely value of priority. #Based from the top 10 callTypes above, came up with the priority values listed in cPrio below. The for loop fixed the missing priority for the top 10 callTypes.

```
##6b2 Empty/NaN/NULL
#Now, set those priority Null values to those observed above.
cList = ['COMMON ASSAULT','BURGLARY','DISORDERLY','ARMED PERSON','AGGRAV ASSAULT',
            'LARCENCY','DESTRUCT PROPTY','DISCHRG FIREARM','FAMILY DISTURB','HIT AND RUN']
cPrio = ['Medium','Medium','Medium', 'High','High',
         'Low','Low', 'High','Medium','Low']
for ii in range(10):
    rx = (df911.callType == cList[ii]) & df911.priority.isnull()
    df911.loc[rx, 'priority'] = cPrio[ii]
```

```
dfPnull = df911[df911.priority.isnull()]
dfPnull.callType.value_counts().head(10)
```

```
SUSPICIOUS PERS     102
ROBBERY ARMED        98
NARCOTICSOutside     94
SILENT ALARM         86
AUTO THEFT           72
AUTO ACCIDENT        70
LARCENY F/AUTO       62
MENTAL CASE          53
JUV DISTURBANCE      49
DWI                  45
Name: callType, dtype: int64
```

```
##6c Uppercase/Lowercase, extra spaces, wrong spelling - make all CallTypes Upper() #CHK RE
df911.callType = df911.callType.str.upper()

##6d Mistyped, extra spaces,... data error ()
df911.callType = df911.callType.str.replace("LARCENCY", "LARCENY") #
df911.callType = df911.callType.str.replace("  ", " ") #608034, 237, 7 >>> 3x
df911.callType = df911.callType.str.replace("  ", " ") #608034, 237, 7 >>> 3x
df911.callType = df911.callType.str.replace("  ", " ") #608034, 237, 7 >>> 3x
```

```
##6d Data error (priority = ['Out of Service', 'Emergency']) #
df911 = df911[df911.priority !='Out of Service']        # need to drop this priority r=880
df911.priority.replace('Emergency','High',inplace=True) # re-tag as High. r=492913+1120 =49
```

Base df = df911;The following dataframes also serve as the basis for each categorical data.
##6e dfDistrict. Top 5 district only: [01/2015, 01/2016, 01/2017]=highest priority call/dist
##6f dfPriority. Top 4 priority only: [01/2015 - 12/2017]
##6g dfCallType. Top 3 callType only: [01/2015, 01/2016, 01/2017]

```
##6e dfDistrict. Top 5 district: [01/2015, 01/2016, 01/2017] = highest priority call/district
df9115 = df911.loc[(df911.dT >= '2015-01') & (df911.dT < '2015-02')] # 83994;
df9116 = df911.loc[(df911.dT >= '2016-01') & (df911.dT < '2016-02')] # 80087;
df9117 = df911.loc[(df911.dT >= '2017-01') & (df911.dT < '2017-02')] # 84422;
df9117.shape #chk (84422, 7), priority 3122560/3123855
```

```
(84404, 7)
```

```
dfCallTypeDist = pd.concat([df9115, df9116, df9117]) # 248503 [01/2015, 01/2016, 01/2017]

dfCallType = pd.DataFrame(dfCallTypeDist, columns=['callType', 'priority', 'district','yr', 'dy'])

dfDistrict = pd.DataFrame(dfCallTypeDist, columns=['priority', 'district','yr', 'dy'])
dfDistrict =dfDistrict[dfDistrict.priority =='High'] #ONLY need HIGH. dfDistrict=BASIS
dfDistrict.head() #chk
```

	priority	district	yr	dy
1106565	High	NW	2015	1
357881	High	SW	2015	1

##6f Work only on top 4 priority for 3 years: [01/2015 - 12/2017], priority district callType dT yr mo dy
dfPriority.priority.value_counts()

##6f Need to make a decision here:
###1) Drop all priority='Out of Service', (r = 880)
###2) Move priority 'Emergency' to the 'High' bin. (r = 1120)
dfPriority = dfPriority[dfPriority.priority !='Out of Service'] # <END dfPriority wrangling>

```
##6f ###2) Move priority 'Emergency' to the 'High' bin.
dfPriority.priority.replace('Emergency', 'High', inplace=True)
dfPriority.priority.value_counts()
```

```
Medium           1559689
Low               706512
High              494883
Non-Emergency     360596
Name: priority, dtype: int64
```

##6g callType. Work only on top 5 callType
###At this point, df911 has been wrangled already on the callType values which affected the priority totals. So, from 6e, # <END dfCallType wrangling>

```
ctList = ['911/NO VOICE','TRAFFIC STOP','DISORDERLY','COMMON ASSAULT','AUTO ACCIDENT']
dfCallType = dfCallType[dfCallType.callType.isin(ctList)] #shape=(103934, 5)
```

```
dfCallType.callType.value_counts()  #chk, just the top 5
```

```
911/NO VOICE      42105
TRAFFIC STOP      22486
DISORDERLY        19106
COMMON ASSAULT    10363
AUTO ACCIDENT      9852
Name: callType, dtype: int64
```

Conclusion/Findings.

Following are the questions that the author wanted to answer based from the dataset, the Python codes used, and the visualization and charts that provide answers to these questions. The codes and graphs/charts are pulled out from the notebook “Data/ 911PoliceCalls4Svc.EDA.wrangle.ipynb”.

1) What district has the greatest <High> priority calls for (Jan/2015, Jan/2016, Jan/2017)?

a) Code. Dataframe: dfDistrict

```
##6e By observation, there are 4 districts ['CW','INFO','SS','TRU'] with insignificant values.
#Remove the lowest 4 districts from the set and work on the top 9 districts.
pd.crosstab(dfDistrict.yr, dfDistrict.district, rownames=['Jan/Year'],
        colnames=['District'], margins=True, margins_name='<TOTAL>')
```

```
# Most <High> Priority Calls/District for 3 Years on the Month of January
dfDistrict = dfDistrict[~dfDistrict.district.isin(['CW','INFO','SS','TRU'])]
dfCT = pd.crosstab(dfDistrict.yr, dfDistrict.district, rownames=['Jan/Year'],
        colnames=['District'], margins=True, margins_name='<TOTAL>')
dfCT
```

b) **Visualization / Chart.**

District Jan/Year	CD	ED	ND	NE	NW	SD	SE	SW	WD	<TOTAL>
2015	714	639	966	1150	813	823	806	758	607	7276
2016	1567	1404	1509	2447	1335	1595	2122	1335	1400	14714
2017	1589	1414	1478	2221	1320	1872	2251	1728	1870	15743
<TOTAL>	3870	3457	3953	5818	3468	4290	5179	3821	3877	37733

Based from the district charts above, we can observe that:
NE (1150) has the highest High priority calls for year 2015, as well as
for 2016 (2447). Whereas, SE (2251) has the highest High priority calls for year 2017.
Overall, NE (5818) has the highest High priority calls for all 3 years.

2) What are the top 4 priority calls for 3 years (2015-2017)?

a) Code. Dataframe: dfPriority

```
dfPriority.priority.value_counts()
Medium            1561149
Low                705902
High               492913
Non-Emergency      360596
Emergency            1120
Out of Service        880
```

```
##6f Need to make a decision here:
###1) Drop all priority='Out of Service',
###2) Move priority 'Emergency' to the 'High' bin.

dfPriority = dfPriority[dfPriority.priority !='Out of Service']  # r=880
dfPriority.priority.value_counts()
```

```
##6f ###2) Move priority 'Emergency' to the 'High' bin.
dfPriority.priority.replace('Emergency', 'High', inplace=True)
dfPriority.priority.value_counts()
```

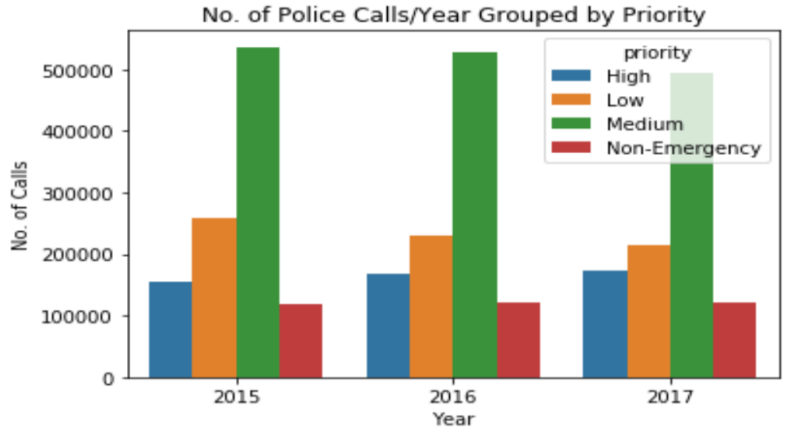
```
Medium           1559689
Low               706512
High              494883
Non-Emergency     360596
Name: priority, dtype: int64
```

```
##6f Work only on top 4 priority: [01/2015 - 12/2017], crosstab
ct = pd.crosstab(dfPriority.yr, dfPriority.priority)
tt = 'No. of Police Calls/Year Grouped by Priority'
stacked = ct.stack().reset_index().rename(columns={0:'val'})
bp = sns.barplot(x=stacked.yr, y=stacked.val, hue=stacked.priority)
bp.set(xlabel='Year', ylabel='No. of Calls', title=tt)
```

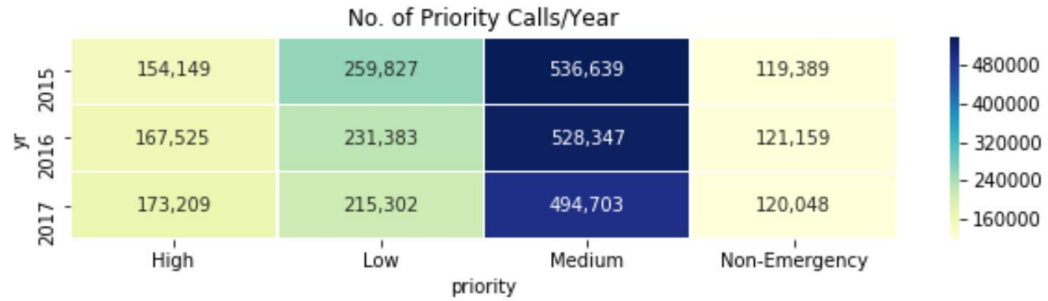
```
tt = 'Percentage of Priority Calls/Year'
ct = pd.crosstab(index=[dfPriority.yr], columns=dfPriority.priority,
            rownames=['Year'], colnames=['Priority'])
plt.figure(figsize=(10,2))
plt.title(tt)

ax = sns.heatmap(pd.crosstab([dfPriority.yr], [dfPriority.priority], normalize='index')
        , yticklabels=1, cmap="YlGnBu", annot=True, cbar=True, linewidths=.5)
```

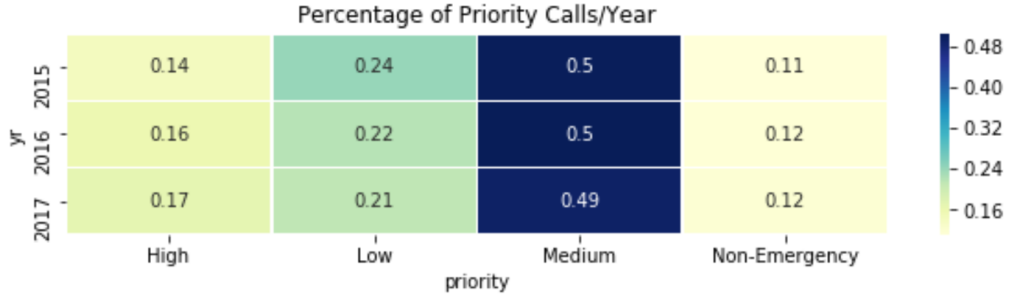
```
tt = 'No. of Priority Calls/Year'
plt.figure(figsize=(10,2))
plt.title(tt)
ax = sns.heatmap(pd.crosstab([dfPriority.yr], [dfPriority.priority]), yticklabels=1
    , cmap="YlGnBu", annot=True, cbar=True, linewidths=.5, fmt=',.0f', cbar_kws={'format': '%.0f'})
```

b) Visualization / Chart.



The barplot above shows clearly that the medium priority calls are the most calls for the years 2015-2017. Low priority calls come second, High is third and Non-Emergency is the lowest for all three years.





Based from the charts above, we can observe that:

i. The High priority call has the THIRD largest number of calls at 14% (154,149), 16% (167,525), and 17% (173,209) for years 2015, 2016, and 2017, respectively.

ii. The Medium priority call TOPs the number of calls at 50% (536,639), 50% (528,347), and 49% (494,703) for years 2015, 2016, and 2017, respectively.

iii. The Low priority calls has the SECOND largest number of calls at 24% (259,827), 22% (231,383), and 21% (215,302) for years 2015, 2016, and 2017, respectively.

iv. The Non-Emergency calls has the LOWEST number of calls at 11% (119,389), 12% (121,159), and 12% (120,0) for years 2015, 2016, and 2017, respectively.

3) What month(s) did the most police calls occur for each year (2015, 2016, 2017) based on each priority call (High, Medium, Low, and Non-Emergency)?

a. Code. Dataframe: dfPriority

```
df15 = dfPriority[dfPriority.yr==2015]
df15c = pd.crosstab(df15.mo, df15.priority)
df15c['TOTAL'] = df15c.High + df15c.Medium + df15c.Low + df15c['Non-Emergency']

df16 = dfPriority[dfPriority.yr==2016]
df16c = pd.crosstab(df16.mo, df16.priority)
df16c['TOTAL'] = df16c.High + df16c.Medium + df16c.Low + df16c['Non-Emergency']

df17 = dfPriority[dfPriority.yr==2017]
df17c = pd.crosstab(df17.mo, df17.priority)
df17c['TOTAL'] = df17c.High + df17c.Medium + df17c.Low + df17c['Non-Emergency']
```

```
#Monthly Calls/Year Grouped by Priority
#================================================
plt.style.use('ggplot')
plt.figure(figsize=(16,12))

#High Priority Calls
#================================================
plt.subplot(2,2,1)
plt.title('Monthly High Priority Calls/Year')
plt.plot(df15c.index, df15c.High, color='red', marker='o', label='2015')
plt.plot(df16c.index, df16c.High, color='green', marker='o', label='2016')
plt.plot(df17c.index, df17c.High, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('No. Of Calls')
plt.xlim(1, 12)
plt.legend(loc='upper right')
```

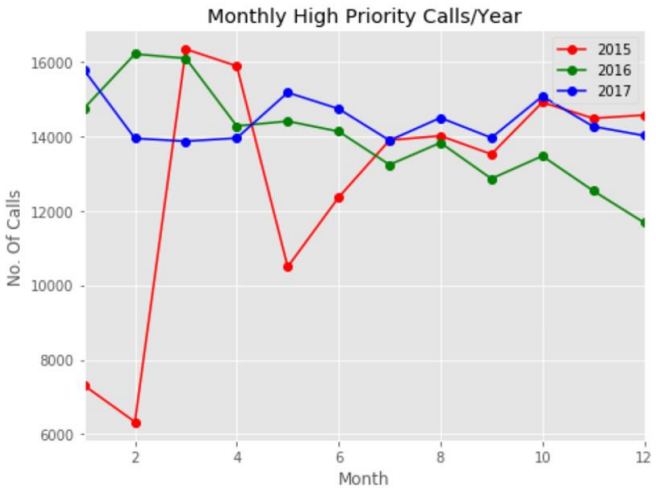
```
#Medium Priority Calls
#================================================
plt.subplot(2, 2, 2)
plt.title('Monthly Medium Priority Calls/Year')
plt.plot(df15c.index, df15c.Medium, color='red', marker='o', label='2015')
plt.plot(df16c.index, df16c.Medium, color='green', marker='o', label='2016')
plt.plot(df17c.index, df17c.Medium, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.xlim(1, 12)
plt.legend(loc='upper right')
```

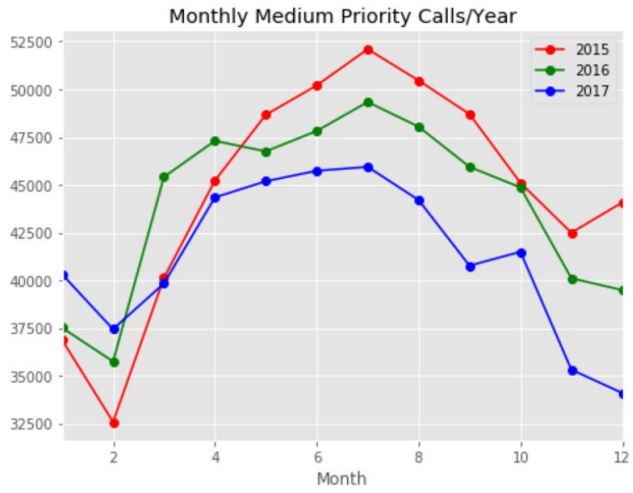
```
#Low Priority Calls
#================================================
plt.subplot(2, 2, 3)
plt.title('Monthly Low Priority Calls/Year')
plt.plot(df15c.index, df15c.Low, color='red', marker='o', label='2015')
plt.plot(df16c.index, df16c.Low, color='green', marker='o', label='2016')
plt.plot(df17c.index, df17c.Low, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('No. Of Calls')
plt.xlim(1, 12)
plt.legend(loc='upper right')
```

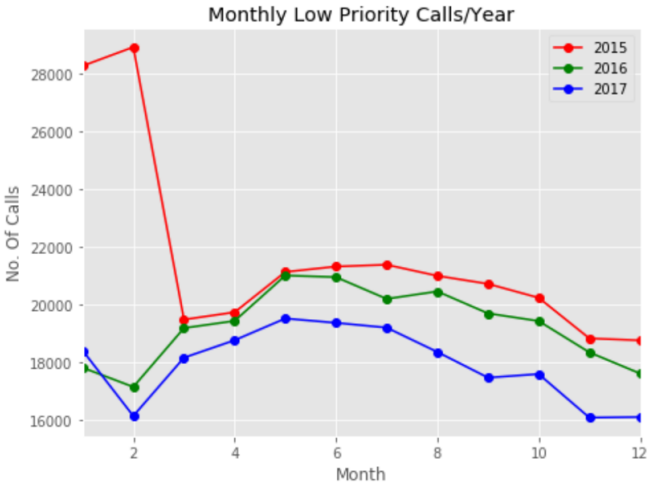
```
#Non-Emergency Calls
#================================================
plt.subplot(2, 2, 4)
plt.title('Monthly Non-Emergency Calls/Year')
plt.plot(df15c.index, df15c['Non-Emergency'], color='red', marker='o', label='2015')
plt.plot(df16c.index, df16c['Non-Emergency'], color='green', marker='o', label='2016')
plt.plot(df17c.index, df17c['Non-Emergency'], color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.xlim(1, 12)
plt.legend(loc='upper right')
#plt.savefig('Data/monthlyCallsByPriority.png')
```

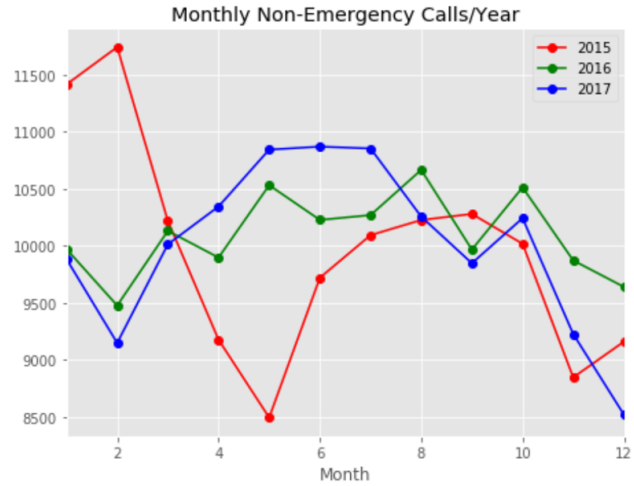
```
#Monthly Total Calls
#================================================
plt.title('Monthly Total Calls/Year')
plt.plot(df15c.index, df15c.TOTAL, color='red', marker='o', label='2015')
plt.plot(df16c.index, df16c.TOTAL, color='green', marker='o', label='2016')
plt.plot(df17c.index, df17c.TOTAL, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('No. Of Calls')
plt.xlim(1, 12)
plt.legend(loc='upper right')
```

b. Visualization/ Chart.









For monthly high priority calls,

i. The most calls occurred in the month of March for the year 2015.
ii. The most calls occurred in the month of February for the year 2016.
iii. The most calls occurred in the month of January for the year 2017.

For monthly medium priority calls,

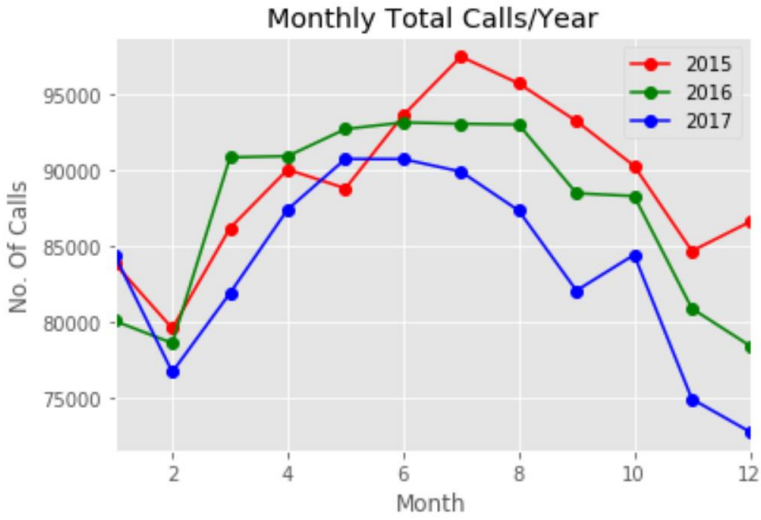
iv. The most calls occurred in the month of July for all three years (2015-2017).

For monthly low priority calls,

v. The most calls occurred in the month of February for the year 2015.
vi. The most calls occurred in the month of May for both years of 2016 and 2017.

For monthly non-emergency calls,

vii. The most calls occurred in the month of February for the year 2015.
viii. The most calls occurred in the month of August for the year 2016.
ix. The most calls occurred in the months of May – July for the year 2017.



For monthly total calls,

x. The most calls occurred in the month of July for the year 2015.

xi. The most calls occurred in the months of June - August for the year 2016.

xii. The most calls occurred in the month of May for the year 2017.

4) What are the top 3 callTypes (type of emergency calls) for (Jan/2015, Jan/2016, and Jan/2017)?

a) Code. Dataframe: dfCallType

```
##from 6e, Work only on top 5 callType: [01/2015, 01/2016, 01/2017], #SUB: #32: Cleaned up f
dfCallType.callType.value_counts().head(5)  #shape=(248503, 5)
```

```
911/NO VOICE      42105
TRAFFIC STOP      22486
DISORDERLY        19106
COMMON ASSAULT    10363
AUTO ACCIDENT      9852
Name: callType, dtype: int64
```

```
ctList = ['911/NO VOICE','TRAFFIC STOP','DISORDERLY','COMMON ASSAULT','AUTO ACCIDENT']
dfCallType = dfCallType[dfCallType.callType.isin(ctList)] #shape=(103934, 5)
```

```
dfCallType.callType.value_counts()  #chk, just the top 5
```

```
ct = pd.crosstab(dfCallType.yr, dfCallType.callType, rownames=['Jan/Year'], colnames=['CallType'])
ct  #chk. Top 5 callType:   <<<END>>> #priority district callType dT yr mo dy.
```

```
tt = 'No. of Police Calls/Year (Month of January) Grouped by CallType'
stacked = ct.stack().reset_index().rename(columns={0:'val'})
bp = sns.barplot(x=stacked['Jan/Year'], y=stacked.val, hue=stacked.CallType)
bp.set(xlabel='Year', ylabel='No. of Calls', title=tt)
```

b) Visualization / Chart.

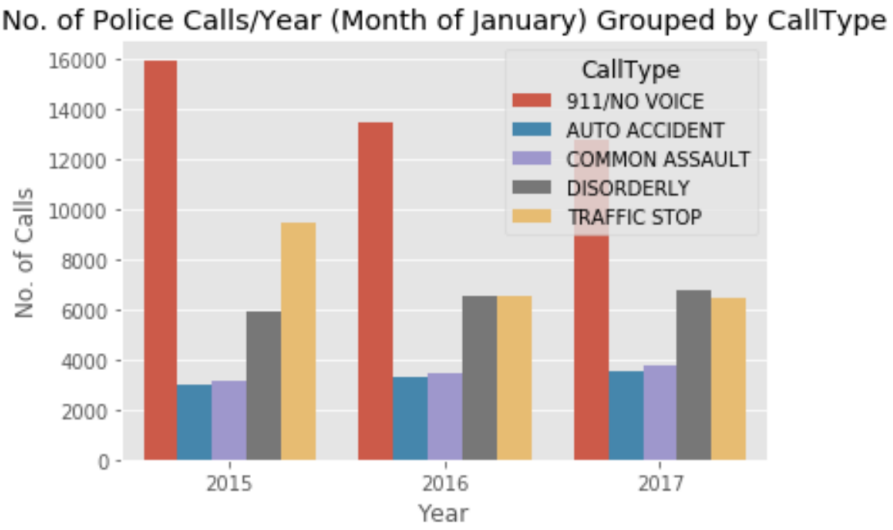
CallType	911/NO VOICE	AUTO ACCIDENT	COMMON ASSAULT	DISORDERLY	TRAFFIC STOP
Jan/Year					
2015	15935	3002	3141	5877	9449
2016	13449	3317	3431	6506	6557
2017	12721	3533	3791	6723	6480

From the callType crosstab chart above:

i. CallType 911/No Voice is the #1 emergency incurred for 3 consecutive years (2015-2017). This cannot be exactly determined what's the reason for the call, except that the caller is not able to talk for whatever reason.

ii. CallType Traffic Stop ranks #2 for 2015 and 2016, and comes #3 in 2017.

iii. CallType Disorderly is the #3 for 2015 and 2016, and #2 in 2017.



The barplot above shows that CallType 911/No Voice is the #1 emergency incurred for 3 consecutive years. Likewise, Traffic Stop appears as rank #2 for 2015, whereas Disorderly appears as rank #2 for 2017. In 2016, Traffic Stop and Disorderly appear to tie for #2.

5) What are the percentages of emergencies calls and non-emergency calls?

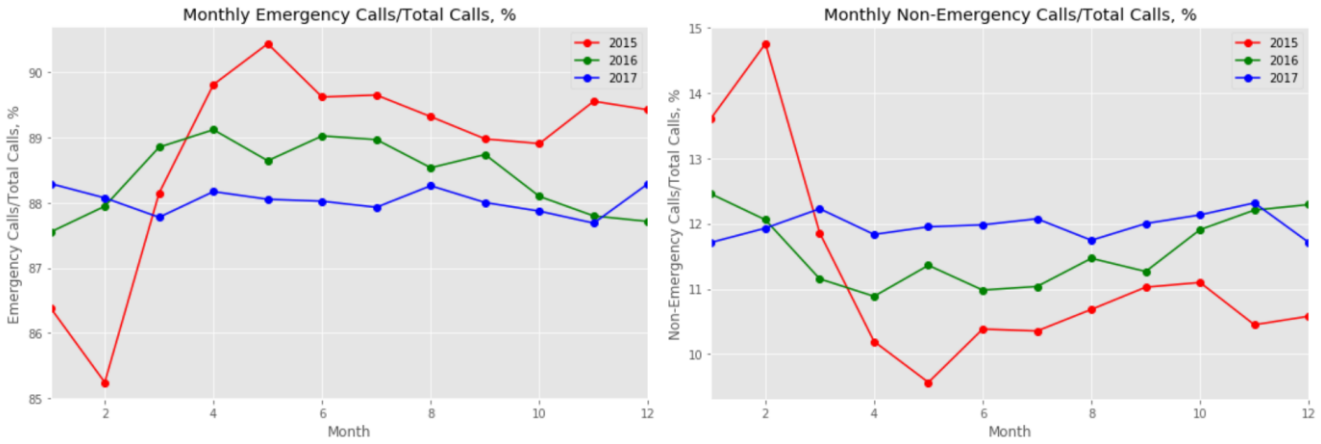
a) Code. Dataframe: dfPriorityPct

Calculate the percentages of Non-Emergency calls and Emergency calls out of the total calls. Total calls = Non-Emergency calls + Emergency calls.

```
#5)*What is the percentage of emergencies and non-emergencies are from a dataset?
#Calc Non-Emergency % and Emergency %, High %, Medium %, and Low %.
# From dfs: crosstab df15c, df16c, df17c; dfPriority[dfPriority.yr==2015]
#df15c = pd.crosstab(df15.mo, df15.priority)
df15p = df15c.copy()
df15p.columns = df15p.columns.str.replace('Non-Emergency','NonEmergency')
df15p['Emergency'] = df15p.High + df15p.Medium + df15p.Low
df15p['EmerPct'] = df15p.Emergency / df15p.TOTAL * 100
df15p['NonEPct'] = df15p.NonEmergency / df15p.TOTAL * 100

df16p = df16c.copy()
df16p.columns = df16p.columns.str.replace('Non-Emergency','NonEmergency')
df16p['Emergency'] = df16p.High + df16p.Medium + df16p.Low
df16p['EmerPct'] = df16p.Emergency / df16p.TOTAL * 100
df16p['NonEPct'] = df16p.NonEmergency / df16p.TOTAL * 100

df17p = df17c.copy()
df17p.columns = df17p.columns.str.replace('Non-Emergency','NonEmergency')
df17p['Emergency'] = df17p.High + df17p.Medium + df17p.Low
df17p['EmerPct'] = df17p.Emergency / df17p.TOTAL * 100
df17p['NonEPct'] = df17p.NonEmergency / df17p.TOTAL * 100
```

```
#5) What are the percentages of Emergency Calls and Non-Emergency calls?
#Monthly Percentage of Emergencies and Non-Emergencies/Year Grouped by Priority
#================================================
plt.style.use('ggplot')
plt.figure(figsize=(16,16))

#Percentage of Emergency Calls
#================================================
plt.subplot(3, 2, 1)
plt.title('Monthly Emergency Calls/Total Calls, %')
plt.plot(df15p.index, df15p.EmerPct, color='red', marker='o', label='2015')
plt.plot(df16p.index, df16p.EmerPct, color='green', marker='o', label='2016')
plt.plot(df17p.index, df17p.EmerPct, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('Emergency Calls/Total Calls, %')
plt.xlim(1, 12)
plt.legend(loc='upper right')

#Percentage of Non-Emergency Calls
#================================================
plt.subplot(3, 2, 2)
plt.title('Monthly Non-Emergency Calls/Total Calls, %')
plt.plot(df15p.index, df15p.NonEPct, color='red', marker='o', label='2015')
plt.plot(df16p.index, df16p.NonEPct, color='green', marker='o', label='2016')
plt.plot(df17p.index, df17p.NonEPct, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('Non-Emergency Calls/Total Calls, %')
plt.xlim(1, 12)
plt.legend(loc='upper right')
```

b) Visualization / Chart.



For 2015, most emergency calls occurred in May (about 90.5%) whereas the same calls occurred in April, June and July in 2016. Interestingly, these calls are likely flat throughout the year in 2017. Since a call is either emergency or non-emergency, the non-emergency calls on the right graph is conversely proportional to the emergency calls on left.

```
df17p
```

priority mo	High	Low	Medium	NonEmergency	TOTAL	Emergency	EmerPct	NonEPct	HighPct	MedPct	LowPct
1	15786	18419	40315	9884	84404	74520	88.289655	11.710345	21.183575	54.099571	24.716855
2	13948	16155	37442	9149	76694	67545	88.070775	11.929225	20.649937	55.432675	23.917388
3	13870	18171	39841	10013	81895	71882	87.773368	12.226632	19.295512	55.425559	25.278929
4	13956	18775	44341	10344	87416	77072	88.166926	11.833074	18.107743	57.531918	24.360338
5	15179	19526	45188	10843	90736	79893	88.049947	11.950053	18.999161	56.560650	24.440189
6	14743	19376	45737	10869	90725	79856	88.019840	11.980160	18.461982	57.274344	24.263675
7	13893	19210	45941	10853	89897	79044	87.927295	12.072705	17.576287	58.120793	24.302920
8	14501	18369	44207	10256	87333	77077	88.256444	11.743556	18.813654	57.354334	23.832012
9	13964	17480	40770	9847	82061	72214	88.000390	11.999610	19.336971	56.457197	24.205833
10	15079	17604	41494	10242	84419	74177	87.867660	12.132340	20.328404	55.939173	23.732424
11	14268	16099	35325	9227	74919	65692	87.684032	12.315968	21.719540	53.773671	24.506789
12	14022	16118	34102	8521	72763	64242	88.289378	11.710622	21.826842	53.083652	25.089505

Looking at the chart for 2017 above, the author observed that:

(1) Both the NonEmergency and Emergency calls (NonEPct and EmerPct) appear to be likely flat from March until October.

(2) There appears to be a surge in December on Emergency calls and a dip in September on NonEmergency calls.

```
df16p
```

priority mo	High	Low	Medium	NonEmergency	TOTAL	Emergency	EmerPct	NonEPct	HighPct	MedPct	LowPct
1	14755	17826	37516	9971	80068	70097	87.546835	12.453165	21.049403	53.520122	25.430475
2	16214	17159	35750	9477	78600	69123	87.942748	12.057252	23.456737	51.719399	24.823865
3	16098	19195	45416	10131	90840	80709	88.847424	11.152576	19.945731	56.271296	23.782973
4	14281	19443	47303	9897	90924	81027	89.115085	10.884915	17.624989	58.379306	23.995705
5	14409	21014	46752	10531	92706	82175	88.640433	11.359567	17.534530	56.893216	25.572254
6	14134	20957	47821	10227	93139	82912	89.019637	10.980363	17.046990	57.676814	25.276196
7	13240	20205	49337	10269	93051	82782	88.964116	11.035884	15.993815	59.598705	24.407480
8	13831	20463	48046	10665	93005	82340	88.532875	11.467125	16.797425	58.350741	24.851834
9	12865	19703	45941	9967	88476	78509	88.734798	11.265202	16.386656	58.516858	25.096486
10	13474	19438	44865	10512	88289	77777	88.093647	11.906353	17.323888	57.684148	24.991964
11	12540	18356	40113	9872	80881	71009	87.794414	12.205586	17.659733	56.490022	25.850244
12	11684	17624	39487	9640	78435	68795	87.709568	12.290432	16.983792	57.398067	25.618141

Based from the 2016 chart above, the author observed that:

(1) Like the 2017 data, both the NonEmergency and Emergency calls (NonEPct and EmerPct) appear to be likely flat from March until October.

(2) There appears also a surge in December on Emergency calls and a little dip in September and April on NonEmergency calls.

```
df15p
```

priority mo	High	Low	Medium	NonEmergency	TOTAL	Emergency	EmerPct	NonEPct	HighPct	MedPct	LowPct
1	7311	28262	36870	11414	83857	72443	86.388733	13.611267	10.092072	50.895187	39.012741
2	6330	28912	32589	11741	79572	67831	85.244810	14.755190	9.332016	48.044404	42.623579
3	16345	19488	40132	10218	86183	75965	88.143833	11.856167	21.516488	52.829593	25.653920
4	15892	19743	45225	9178	90038	80860	89.806526	10.193474	19.653722	55.930002	24.416275
5	10497	21136	48668	8495	88796	80301	90.433128	9.566872	13.072066	60.606966	26.320967
6	12361	21323	50206	9720	93610	83890	89.616494	10.383506	14.734772	59.847419	25.417809
7	13899	21385	52104	10093	97481	87388	89.646187	10.353813	15.904930	59.623747	24.471323
8	14015	21002	50465	10226	95708	85482	89.315418	10.684582	16.395265	59.035820	24.568915
9	13523	20724	48706	10279	93232	82953	88.974816	11.025184	16.302002	58.715176	24.982822
10	14915	20243	45101	10018	90277	80259	88.903043	11.096957	18.583586	56.194321	25.222093
11	14487	18839	42495	8847	84668	75821	89.550952	10.449048	19.106844	56.046478	24.846678
12	14574	18770	44078	9160	86582	77422	89.420434	10.579566	18.824107	56.932138	24.243755

(1) Unlike the 2017/2016 data, the 2015 NonEmergency calls exhibit a dip on April to June and November to December, whereas the Emergency calls shows a dip on February.

6) What are the percentages of High, Medium, and Low priorities from the emergency calls?

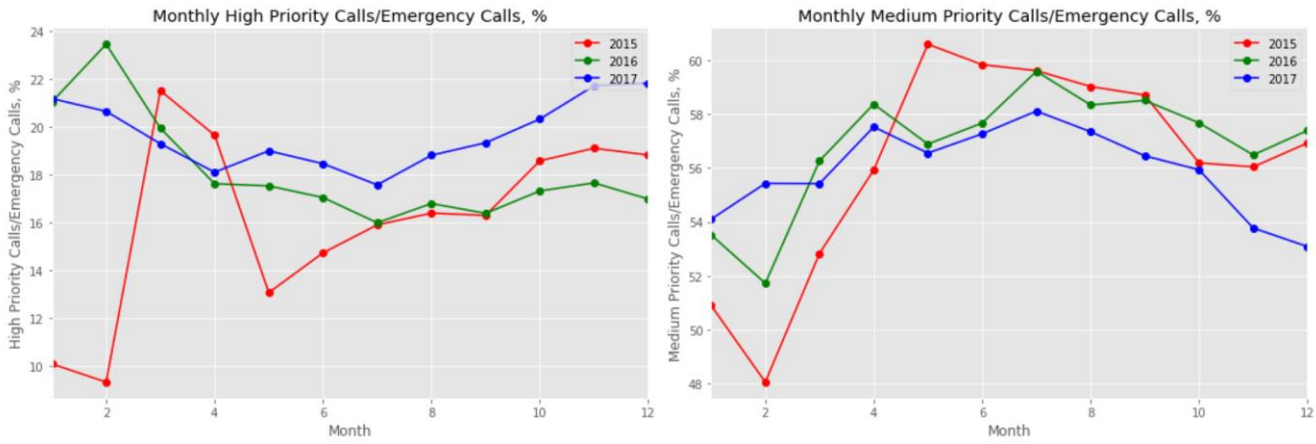
a) Code. Dataframe:

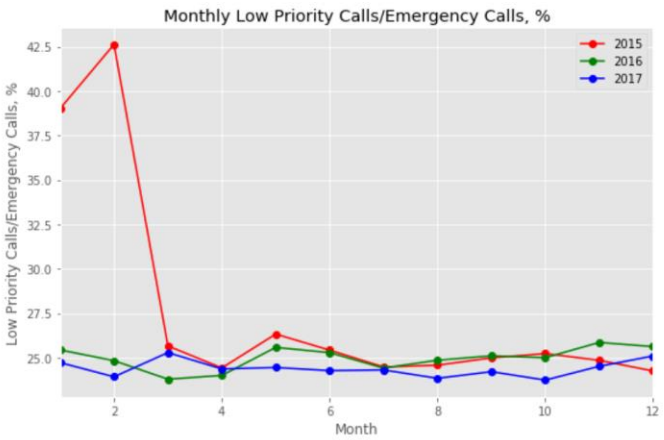
Based from the dataset above, calculate the percentages of High, Medium, and Low priority calls out of the total emergency calls.

```
df15p['HighPct'] = df15p.High / df15p.Emergency * 100
df15p['MedPct'] = df15p.Medium / df15p.Emergency * 100
df15p['LowPct'] = df15p.Low / df15p.Emergency * 100

df16p['HighPct'] = df16p.High / df16p.Emergency * 100
df16p['MedPct'] = df16p.Medium / df16p.Emergency * 100
df16p['LowPct'] = df16p.Low / df16p.Emergency * 100

df17p['HighPct'] = df17p.High / df17p.Emergency * 100
df17p['MedPct'] = df17p.Medium / df17p.Emergency * 100
df17p['LowPct'] = df17p.Low / df17p.Emergency * 100
```

```
#Percentage of High Priority Calls
#================================================
plt.subplot(3, 2, 3)
plt.title('Monthly High Priority Calls/Emergency Calls, %')
plt.plot(df15p.index, df15p.HighPct, color='red', marker='o', label='2015')
plt.plot(df16p.index, df16p.HighPct, color='green', marker='o', label='2016')
plt.plot(df17p.index, df17p.HighPct, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('High Priority Calls/Emergency Calls, %)
plt.xlim(1, 12)
plt.legend(loc='upper right')

#Percentage of Medium Priority Calls
#================================================
plt.subplot(3, 2, 4)
plt.title('Monthly Medium Priority Calls/Emergency Calls, %')
plt.plot(df15p.index, df15p.MedPct, color='red', marker='o', label='2015')
plt.plot(df16p.index, df16p.MedPct, color='green', marker='o', label='2016')
plt.plot(df17p.index, df17p.MedPct, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('Medium Priority Calls/Emergency Calls, %)
plt.xlim(1, 12)
plt.legend(loc='upper right')

#Percentage of Low Priority Calls
#================================================
plt.subplot(3, 2, 5)
plt.title('Monthly Low Priority Calls/Emergency Calls, %')
plt.plot(df15p.index, df15p.LowPct, color='red', marker='o', label='2015')
plt.plot(df16p.index, df16p.LowPct, color='green', marker='o', label='2016')
plt.plot(df17p.index, df17p.LowPct, color='blue', marker='o', label='2017')
plt.xlabel('Month')
plt.ylabel('Low Priority Calls/Emergency Calls, %)
plt.xlim(1, 12)
plt.legend(loc='upper right')
plt.tight_layout()
```

b) Visualization / Chart.





High Priority Calls:

1. In 2015, most calls occurred in March (almost 22 %), whereas in 2016, most occurred in February (over 23 %). In 2017, most calls occurred in December (about 22 %).

   Medium Priority Calls:

2. In 2015, most calls occurred in May (over 60 %), whereas in 2016 and 2017, most occurred in July, almost 60 % and about 58 %, respectively.

   Low Priority Calls:

3. In 2015, most calls occurred in February (42 %), whereas in 2016, most occurred in May (over 25 %). In 2017, most calls occurred in November (over 25 %).

   An emergency call is either high, medium, or low priority. So, an increase in one priority may decrease one of the other two priorities or both. Conversely, a decrease in one priority may increase one of the other two priorities or both.

priority mo	High	Low	Medium	NonEmergency	TOTAL	Emergency	EmerPct	NonEPct	HighPct	MedPct	LowPct
1	15786	18419	40315	9884	84404	74520	88.289655	11.710345	21.183575	54.099571	24.716855
2	13948	16155	37442	9149	76694	67545	88.070775	11.929225	20.649937	55.432675	23.917388
3	13870	18171	39841	10013	81895	71882	87.773368	12.226632	19.295512	55.425559	25.278929
4	13956	18775	44341	10344	87416	77072	88.166926	11.833074	18.107743	57.531918	24.360338
5	15179	19526	45188	10843	90736	79893	88.049947	11.950053	18.999161	56.560650	24.440189
6	14743	19376	45737	10869	90725	79856	88.019840	11.980160	18.461982	57.274344	24.263675
7	13893	19210	45941	10853	89897	79044	87.927295	12.072705	17.576287	58.120793	24.302920
8	14501	18369	44207	10256	87333	77077	88.256444	11.743556	18.813654	57.354334	23.832012
9	13964	17480	40770	9847	82061	72214	88.000390	11.999610	19.336971	56.457197	24.205833
10	15079	17604	41494	10242	84419	74177	87.867660	12.132340	20.328404	55.939173	23.732424
11	14268	16099	35325	9227	74919	65692	87.684032	12.315968	21.719540	53.773671	24.506789
12	14022	16118	34102	8521	72763	64242	88.289378	11.710622	21.826842	53.083652	25.089505

Looking at the chart for 2017 above, the author observed that:

(1) The High priority call percentage (HighPct) has the least calls from 17.6 % (July) to 21.8 % (December).

(2) The Medium priority call percentage (MediumPct) TOPs calls from 53.1 % (December) to 58.1 % (July).

(3) The Low priority call percentage (LowPct) has the SECOND most calls from 23.7 % (October) to 25.3 % (March).

Recommendations.

1. I'd be interested to continue to work on this dataset to predict the following, among other things, in the future:
    a. What type of emergency calls in a particular time of year in a certain district/location
    b. How many emergency calls () and non-emergency calls in a particular time of year in a certain district/location
    c. What call priority will be the most in a particular time of year in a certain district/location

Appendix

Data Source:
https://data.baltimorecity.gov/Public-Safety/911-Police-Calls-for-Service/xviu-ezkt

Data Columns:
(Rows = 3.86 mil; Columns=8; Each row is a Call)

Column	Description	Type	Comment
recordId	System generated?	number	2749202
callDateTime		datetime	'08/10/2017 03:25:00 PM' format: mm/dd/yyyy hh:nn:ss AM/PM
priority	Priority of emergencies	text	[Low, Medium, High]
district	District codes	text	[ND, NE, SE,…]
description	District Description	text	LARCENY
callNumber	"how is generated? Where's the source?	text	P172221666
incidentLocation		text	3100 ST PAUL ST
Location	IncidentLocation + lat/long	Location (lat/long)	"3100 ST PAUL ST BALTIMORE, MD (39.326142, -76.615959)"

Sample Data:

recordId,callDateTime,priority,district,description,callNumber,incidentLocation,location
2749202,08/10/2017 03:25:00 PM,Low,ND,LARCENY,P172221666,3100 ST PAUL ST,"3100 ST PAUL ST
BALTIMORE, MD (39.326142, -76.615959)"
2785533,08/23/2017 12:53:00 PM,Medium,NE,911/NO VOICE,P172351302,3300 BELAIR RD,"3300 BELAIR RD
BALTIMORE, MD (39.321622, -76.573597)"

recordId	callDateTime	priority	district	description	callNum...	incident...	location
4054762	2018 Dec 13 10:15:00 AM	Medium	SW	NARCOTICSOutside	P183470735	900 POPLAR...	(39.298174°, -76.665121°)
4054775	2018 Dec 13 10:14:00 AM	Low	ED	HIT AND RUN	P183470730	N PATTERSO...	
4054758	2018 Dec 13 10:13:00 AM	Medium	SE	FAMILY DISTURB	P183470732	300 BALLOU...	(39.28776°, -76.59726°)
4054773	2018 Dec 13 10:13:00 AM	Non-Emergency	NW	Court	P183470727	CIRCUIT COU...	(38.802502°, -76.537201°)
4054774	2018 Dec 13 10:12:00 AM	High	NW	RA POLICE	P183470725	4500 KATHLA...	(39.327058°, -76.69471°)
4054754	2018 Dec 13 10:11:00 AM	High	NW	NONBREATHING / A	P183470720	4500 KATHLA...	(39.327058°, -76.69471°)
4054756	2018 Dec 13 10:11:00 AM	Non-Emergency	NE	3P	P183470731	3500 ERDMA...	(39.318585°, -76.570381°)
4054764	2018 Dec 13 10:11:00 AM	Medium	SD	NARCOTICSOutside	P183470724	100 S ARLING...	(39.286577°, -76.635164°)
4054766	2018 Dec 13 10:10:00 AM	Medium	SD	BURGLARY	P183470726	1300 BELT ST	(39.275101°, -76.606152°)
4054760	2018 Dec 13 10:09:00 AM	Medium	SD	SUSPICIOUS PERS	P183470721	BOYLE ST/E H...	
4054761	2018 Dec 13 10:09:00 AM	High	NE	BY THREAT	P183470723	4400 CHALET...	(39.326707°, -76.560482°)
4054767	2018 Dec 13 10:08:00 AM	Low	TRU	AUTO THEFT	P183470760	4200 FALLS RD	(39.33941°, -76.638616°)

References

1) https://www.policedatainitiative.org/datasets/calls-for-service/
2) http://www.ecats911.com

Glossary

PSAP – Public-Safety Answering Point

NENA – National Emergency Number Association

CAD – Computer-Aided Dispatch Systems