

# 911 Emergency Call Analysis

Soumik Chaudhuri  
E-mail: [contact@soumik.in](mailto:contact@soumik.in)

 Soumik.in

# Abstract

Analyze the 911 emergency calls and try to find if the types of accidents vary with every day of the week.

For example, is there a drop in the number of accidents on Sundays when people do not go to their offices mostly?

**Methods Used:** numpy, pandas, matplotlib, seaborn

**Conclusion:** The number of vehicle accidents drops on Sundays.

# Motivation

I live in India where thousands of people die every day on road accidents. If there are lower number of accidents on Sundays, then we can save lives if we work from home whenever possible.

# Dataset

**Dataset:** <https://www.kaggle.com/mchirico/montcoalert>

**Emergency - 911 Calls** (Montgomery County, PA)

# Data Preparation and Cleaning

After exploring the data, we converted the timestamp object type from “string” to “Pandas Timestamp”.

We also extracted three keywords from the title column: EMS, Fire, Traffic

# Research Questions

Does the number of road accidents drop on Sundays?

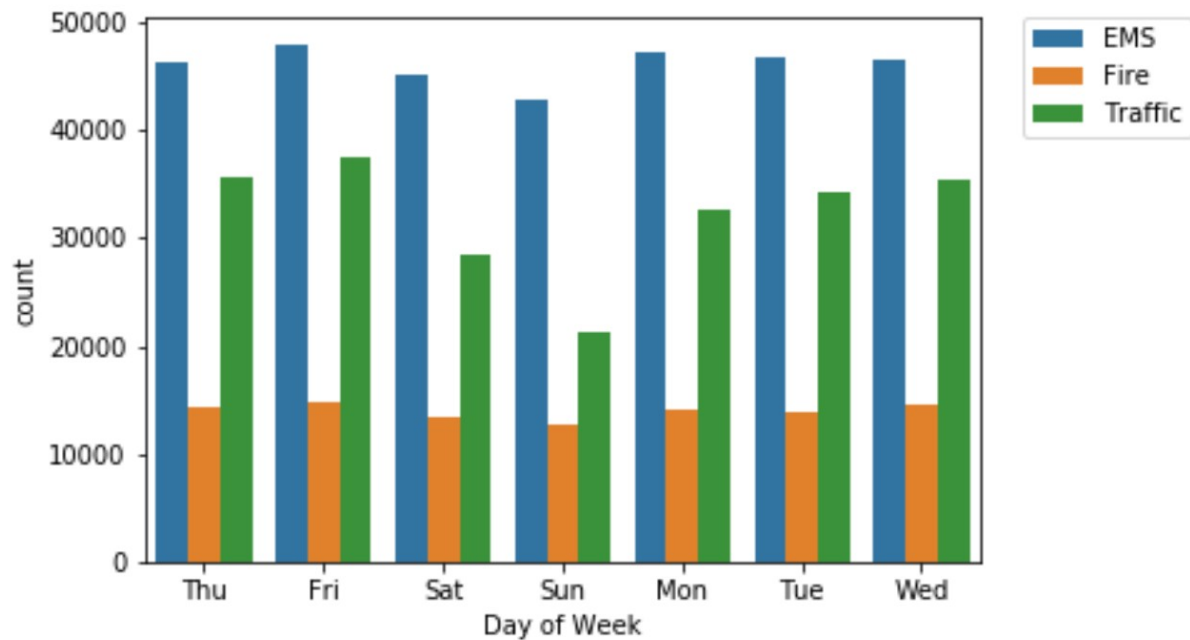
Does the number of EMS cases vary with time?

# Methods

We used seaborn to plot the data.

It helps to visualize the data better.

# Findings





# Conclusion

The number of 911 calls related to "**Fire**" does not vary with the day of the week.

The number of **vehicle accidents** drops on Sunday, which is understood because people do not go to offices on weekends.

# 911 Call Analysis

June 13, 2020

## 911 Call Analysis

**Dataset:** <https://www.kaggle.com/mchirico/montcoalert>

Emergency - 911 Calls (Montgomery County, PA)

**Visualize the type of 911 calls:** EMS, Fire, Vehicle Accident

**Conclusion:** The number of 911 calls related to “Fire” does not vary with the day of the week. The number of vehicle accidents drops on Sunday, which is understood because people do not go to offices on weekends.

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

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

```
[3]: #Explore the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 645716 entries, 0 to 645715
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   lat         645716 non-null  float64
1   lng         645716 non-null  float64
2   desc        645716 non-null  object
3   zip         567493 non-null  float64
4   title       645716 non-null  object
5   timeStamp  645716 non-null  object
6   twp         645434 non-null  object
7   addr        645716 non-null  object
8   e           645716 non-null  int64
dtypes: float64(3), int64(1), object(5)
memory usage: 44.3+ MB
```

```
[4]: df.head(10)
```

```
[4]:      lat      lng      desc \
0  40.297876 -75.581294 REINDEER CT & DEAD END; NEW HANOVER; Station ...
1  40.258061 -75.264680 BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...
2  40.121182 -75.351975 HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...
3  40.116153 -75.343513 AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...
4  40.251492 -75.603350 CHERRYWOOD CT & DEAD END; LOWER POTTS GROVE; S...
5  40.253473 -75.283245 CANNON AVE & W 9TH ST; LANSDALE; Station 345;...
6  40.182111 -75.127795 LAUREL AVE & OAKDALE AVE; HORSHAM; Station 35...
7  40.217286 -75.405182 COLLEGEVILLE RD & LYWISKI RD; SKIPPACK; Stati...
8  40.289027 -75.399590 MAIN ST & OLD SUMNEYTOWN PIKE; LOWER SALFORD;...
9  40.102398 -75.291458 BLUEROUTE & RAMP I476 NB TO CHEMICAL RD; PLYM...
```

```
      zip      title      timeStamp \
0  19525.0      EMS: BACK PAINS/INJURY  2015-12-10 17:10:52
1  19446.0      EMS: DIABETIC EMERGENCY  2015-12-10 17:29:21
2  19401.0      Fire: GAS-ODOR/LEAK     2015-12-10 14:39:21
3  19401.0      EMS: CARDIAC EMERGENCY  2015-12-10 16:47:36
4      NaN      EMS: DIZZINESS         2015-12-10 16:56:52
5  19446.0      EMS: HEAD INJURY       2015-12-10 15:39:04
6  19044.0      EMS: NAUSEA/VOMITING    2015-12-10 16:46:48
7  19426.0      EMS: RESPIRATORY EMERGENCY 2015-12-10 16:17:05
8  19438.0      EMS: SYNCOPAL EPISODE   2015-12-10 16:51:42
9  19462.0      Traffic: VEHICLE ACCIDENT - 2015-12-10 17:35:41
```

```
      twp      addr e
0      NEW HANOVER      REINDEER CT & DEAD END 1
1      HATFIELD TOWNSHIP      BRIAR PATH & WHITEMARSH LN 1
2      NORRISTOWN      HAWS AVE 1
3      NORRISTOWN      AIRY ST & SWEDE ST 1
4      LOWER POTTS GROVE      CHERRYWOOD CT & DEAD END 1
5      LANSDALE      CANNON AVE & W 9TH ST 1
6      HORSHAM      LAUREL AVE & OAKDALE AVE 1
7      SKIPPACK      COLLEGEVILLE RD & LYWISKI RD 1
8      LOWER SALFORD      MAIN ST & OLD SUMNEYTOWN PIKE 1
9      PLYMOUTH      BLUEROUTE & RAMP I476 NB TO CHEMICAL RD 1
```

```
[5]: #Top 5 zip codes for 911 calls
df['zip'].value_counts().head(10)
```

```
[5]: 19401.0    44247
19464.0    42630
19403.0    33934
19446.0    31323
19406.0    21865
19002.0    20566
```

```
19468.0    18480
19046.0    17441
19454.0    17200
19090.0    16927
Name: zip, dtype: int64
```

```
[6]: #Top 5 townships for 911 calls
df['twp'].value_counts().head(10)
```

```
[6]: LOWER MERION          54204
      ABINGTON             38929
      NORRISTOWN           36471
      UPPER MERION         35098
      CHELTENHAM           29774
      POTTSTOWN            26545
      UPPER MORELAND       22276
      LOWER PROVIDENCE     21837
      PLYMOUTH             19533
      UPPER DUBLIN         18433
Name: twp, dtype: int64
```

In the titles column there are “Reasons” specified before the title code. These are EMS, Fire, and Traffic. We will create a new column called “Reason” that contains the string value.

```
[7]: x = df['title'].iloc[0]
      x.split(':')[0]
```

```
[7]: 'EMS'
```

```
[8]: df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])
df['Reason'].head(10)
```

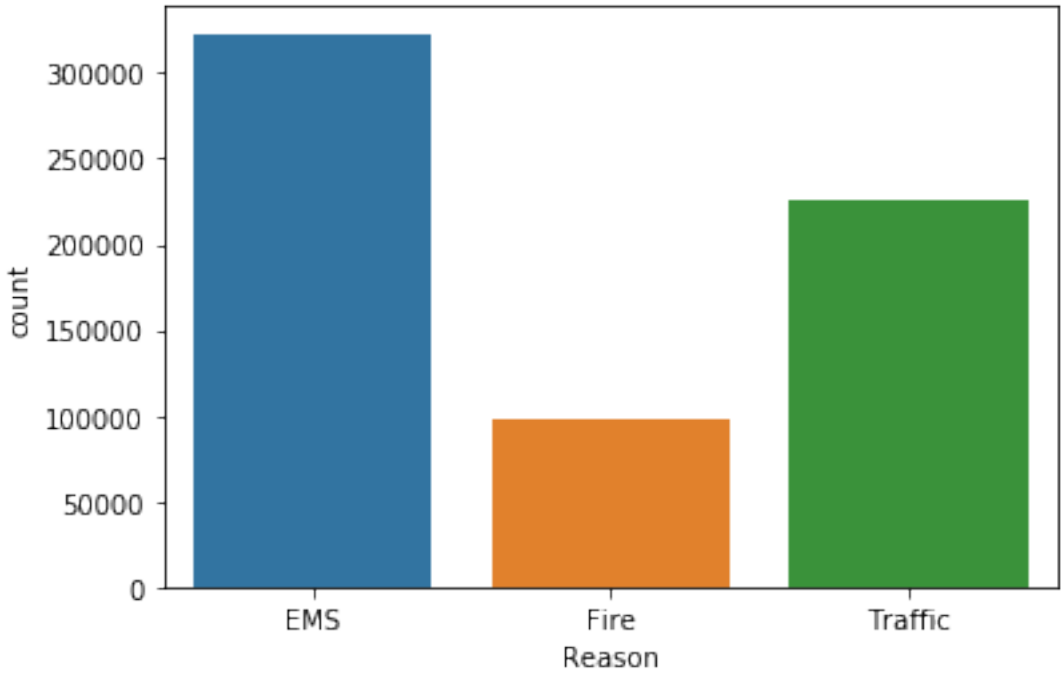
```
[8]: 0      EMS
      1      EMS
      2    Fire
      3      EMS
      4      EMS
      5      EMS
      6      EMS
      7      EMS
      8      EMS
      9    Traffic
Name: Reason, dtype: object
```

```
[9]: #Count the total number of calls for each reason
df['Reason'].value_counts()
```

```
[9]: EMS          322880
      Traffic     225213
      Fire        97623
      Name: Reason, dtype: int64
```

```
[10]: #Create a plot
      sns.countplot(x='Reason', data=df)
```

```
[10]: <matplotlib.axes._subplots.AxesSubplot at 0x14cdfde5608>
```



```
[11]: #Explore the timestamps
      print(df['timeStamp'].iloc[0])
      print()
      print('Object Type: ', end='')
      print(type(df['timeStamp'].iloc[0]))
```

2015-12-10 17:10:52

Object Type: <class 'str'>

```
[12]: #Convert the timestamps to DateTime objects
      df['timeStamp'] = pd.to_datetime(df['timeStamp'])

      type(df['timeStamp'].iloc[0])
```

```
[12]: pandas._libs.tslibs.timestamps.Timestamp
```

```
[13]: df['Hour'] = df['timeStamp'].apply(lambda time: time.hour)
df['Hour'].head(10)
```

```
[13]: 0    17
      1    17
      2    14
      3    16
      4    16
      5    15
      6    16
      7    16
      8    16
      9    17
      Name: Hour, dtype: int64
```

```
[14]: df['Month'] = df['timeStamp'].apply(lambda time: time.month)
df['Day of Week'] = df['timeStamp'].apply(lambda time: time.dayofweek)
df.head(10)
```

```
[14]:      lat      lng      desc \
0  40.297876 -75.581294 REINDEER CT & DEAD END; NEW HANOVER; Station ...
1  40.258061 -75.264680 BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...
2  40.121182 -75.351975 HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...
3  40.116153 -75.343513 AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...
4  40.251492 -75.603350 CHERRYWOOD CT & DEAD END; LOWER POTTS GROVE; S...
5  40.253473 -75.283245 CANNON AVE & W 9TH ST; LANSDALE; Station 345;...
6  40.182111 -75.127795 LAUREL AVE & OAKDALE AVE; HORSHAM; Station 35...
7  40.217286 -75.405182 COLLEGEVILLE RD & LYWISKI RD; SKIPPACK; Stati...
8  40.289027 -75.399590 MAIN ST & OLD SUMNEYTOWN PIKE; LOWER SALFORD;...
9  40.102398 -75.291458 BLUEROUTE & RAMP I476 NB TO CHEMICAL RD; PLYM...
```

```
      zip      title      timeStamp \
0  19525.0      EMS: BACK PAINS/INJURY 2015-12-10 17:10:52
1  19446.0      EMS: DIABETIC EMERGENCY 2015-12-10 17:29:21
2  19401.0      Fire: GAS-ODOR/LEAK 2015-12-10 14:39:21
3  19401.0      EMS: CARDIAC EMERGENCY 2015-12-10 16:47:36
4      NaN      EMS: DIZZINESS 2015-12-10 16:56:52
5  19446.0      EMS: HEAD INJURY 2015-12-10 15:39:04
6  19044.0      EMS: NAUSEA/VOMITING 2015-12-10 16:46:48
7  19426.0      EMS: RESPIRATORY EMERGENCY 2015-12-10 16:17:05
8  19438.0      EMS: SYNCOPAL EPISODE 2015-12-10 16:51:42
9  19462.0      Traffic: VEHICLE ACCIDENT - 2015-12-10 17:35:41
```

```
      twp      addr e      Reason \
0      NEW HANOVER      REINDEER CT & DEAD END 1      EMS
```

1	HATFIELD TOWNSHIP	BRIAR PATH & WHITEMARSH LN	1	EMS
2	NORRISTOWN	HAWS AVE	1	Fire
3	NORRISTOWN	AIRY ST & SWEDE ST	1	EMS
4	LOWER POTTS GROVE	CHERRYWOOD CT & DEAD END	1	EMS
5	LANS DALE	CANNON AVE & W 9TH ST	1	EMS
6	HORSHAM	LAUREL AVE & OAKDALE AVE	1	EMS
7	SKIPPACK	COLLEGEVILLE RD & LYWISKI RD	1	EMS
8	LOWER SALFORD	MAIN ST & OLD SUMNEYTOWN PIKE	1	EMS
9	PLYMOUTH	BLUEROUTE & RAMP I476 NB TO CHEMICAL RD	1	Traffic

	Hour	Month	Day of Week
0	17	12	3
1	17	12	3
2	14	12	3
3	16	12	3
4	16	12	3
5	15	12	3
6	16	12	3
7	16	12	3
8	16	12	3
9	17	12	3

```
[15]: #Map the actual string names to the day of the week
dmap = {0:'Mon', 1:'Tue', 2:'Wed', 3:'Thu', 4:'Fri', 5:'Sat', 6:'Sun'}
df['Day of Week'] = df['Day of Week'].map(dmap)
df.head(10)
```

```
[15]:
```

	lat	lng	desc \
0	40.297876	-75.581294	REINDEER CT & DEAD END; NEW HANOVER; Station ...
1	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...
2	40.121182	-75.351975	HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTS GROVE; S...
5	40.253473	-75.283245	CANNON AVE & W 9TH ST; LANS DALE; Station 345;...
6	40.182111	-75.127795	LAUREL AVE & OAKDALE AVE; HORSHAM; Station 35...
7	40.217286	-75.405182	COLLEGEVILLE RD & LYWISKI RD; SKIPPACK; Stati...
8	40.289027	-75.399590	MAIN ST & OLD SUMNEYTOWN PIKE; LOWER SALFORD;...
9	40.102398	-75.291458	BLUEROUTE & RAMP I476 NB TO CHEMICAL RD; PLYM...

	zip	title	timeStamp \
0	19525.0	EMS: BACK PAINS/INJURY	2015-12-10 17:10:52
1	19446.0	EMS: DIABETIC EMERGENCY	2015-12-10 17:29:21
2	19401.0	Fire: GAS-ODOR/LEAK	2015-12-10 14:39:21
3	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 16:47:36
4	NaN	EMS: DIZZINESS	2015-12-10 16:56:52
5	19446.0	EMS: HEAD INJURY	2015-12-10 15:39:04
6	19044.0	EMS: NAUSEA/VOMITING	2015-12-10 16:46:48

```

7 19426.0 EMS: RESPIRATORY EMERGENCY 2015-12-10 16:17:05
8 19438.0 EMS: SYNCOPAL EPISODE 2015-12-10 16:51:42
9 19462.0 Traffic: VEHICLE ACCIDENT - 2015-12-10 17:35:41

```

	twp	addr	e	Reason	\
0	NEW HANOVER	REINDEER CT & DEAD END	1	EMS	
1	HATFIELD TOWNSHIP	BRIAR PATH & WHITEMARSH LN	1	EMS	
2	NORRISTOWN	HAWS AVE	1	Fire	
3	NORRISTOWN	AIRY ST & SWEDE ST	1	EMS	
4	LOWER POTTSBORO	CHERRYWOOD CT & DEAD END	1	EMS	
5	LANSDALE	CANNON AVE & W 9TH ST	1	EMS	
6	HORSHAM	LAUREL AVE & OAKDALE AVE	1	EMS	
7	SKIPPAK	COLLEGEVILLE RD & LYWISKI RD	1	EMS	
8	LOWER SALFORD	MAIN ST & OLD SUMNEYTOWN PIKE	1	EMS	
9	PLYMOUTH	BLUEROUTE & RAMP I476 NB TO CHEMICAL RD	1	Traffic	

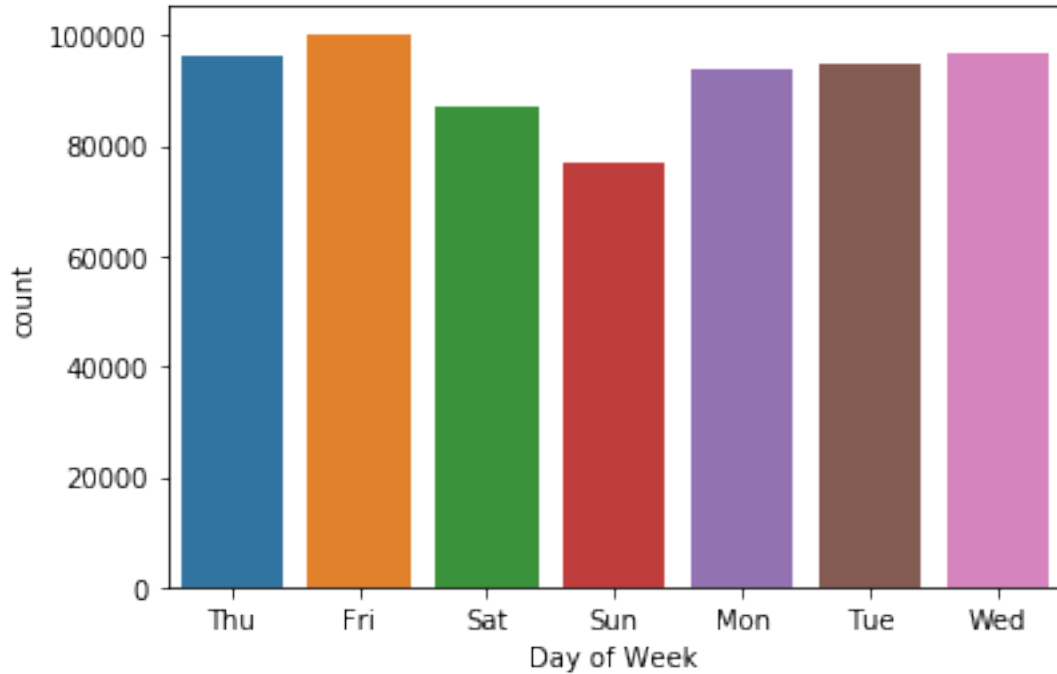
	Hour	Month	Day of Week
0	17	12	Thu
1	17	12	Thu
2	14	12	Thu
3	16	12	Thu
4	16	12	Thu
5	15	12	Thu
6	16	12	Thu
7	16	12	Thu
8	16	12	Thu
9	17	12	Thu

```
[16]: #Create a countplot of the Day of Week based on the Reason column
```

```
sns.countplot(x='Day of Week', data=df)
```

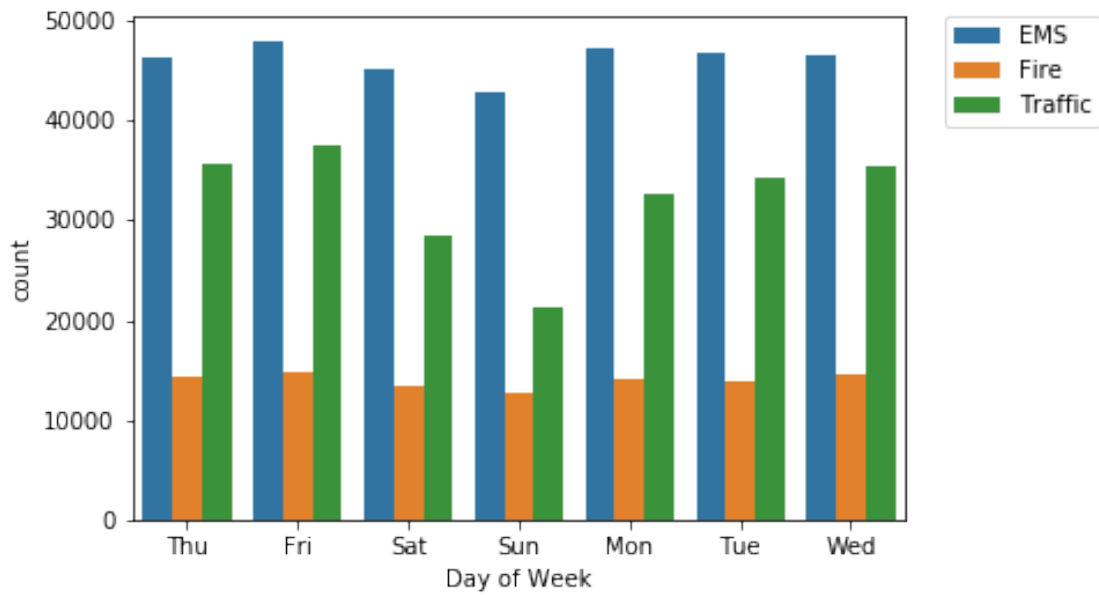
```
[16]: <matplotlib.axes._subplots.AxesSubplot at 0x14cee9fcc48>
```





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

[17]: <matplotlib.legend.Legend at 0x14ccea69048>



[ ]: