# Capstone Project on 911 Call - Montgomery County, PA

For this capstone project we will be analyzing some 911 call data.

Python Version Used: Python 2.7.12 :: Anaconda custom (64-bit)

**Top 5 zip codes for 911 calls:**

**Results:**

19401.0 6979

19464.0 6643

19403.0 4854

19446.0 4748

19406.0 3174

Name: zip, dtype: int64

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

#print df['zip'].value\_counts().head(5)

**Top 5 townships for 911 calls**

**Results:**

LOWER MERION 8443

ABINGTON 5977

NORRISTOWN 5890

UPPER MERION 5227

CHELTENHAM 4575

Name: twp, dtype: int64

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

print df['twp'].value\_counts().head(5)

**Most common Reason for a 911 call based**

Ans: Based on the Title column the data is counted by splitting it into type and creating categories out of it.

EMS 48877

Traffic 35695

Fire 14920

Name: Reason, dtype: int64

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

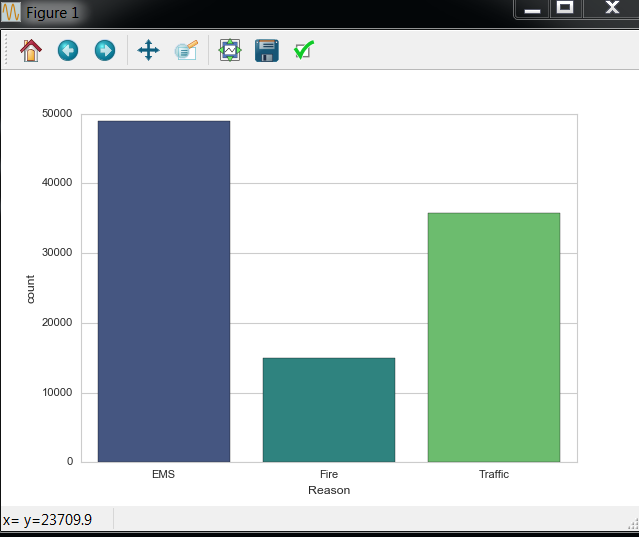
sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

print df['Reason'].value\_counts()

**countplot of 911 calls by Reason.**



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

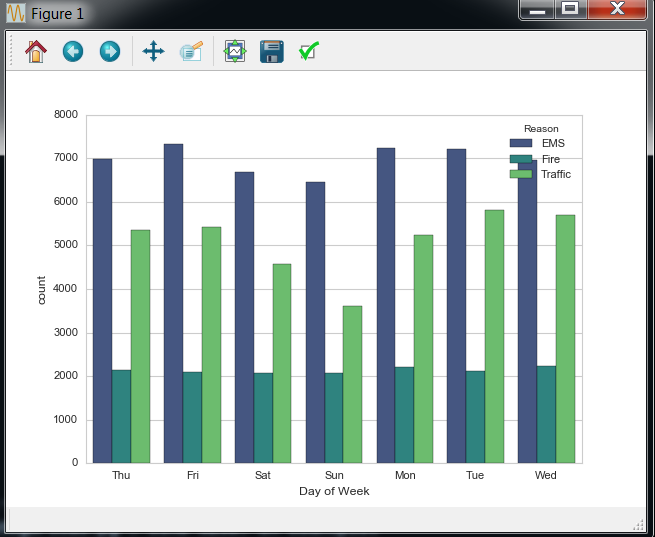
df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

sns.countplot(x='Reason',data=df,palette='viridis')

sns.plt.show()

**No. of calls based on different Reasons depending on Day of the week**

Data Cleaning- The timestamp has been changed from string function to DateTime objects.



Code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

df['timeStamp'] = pd.to\_datetime(df['timeStamp'])

df['Hour'] = df['timeStamp'].apply(lambda time: time.hour)

df['Month'] = df['timeStamp'].apply(lambda time: time.month)

df['Day of Week'] = df['timeStamp'].apply(lambda time: time.dayofweek)

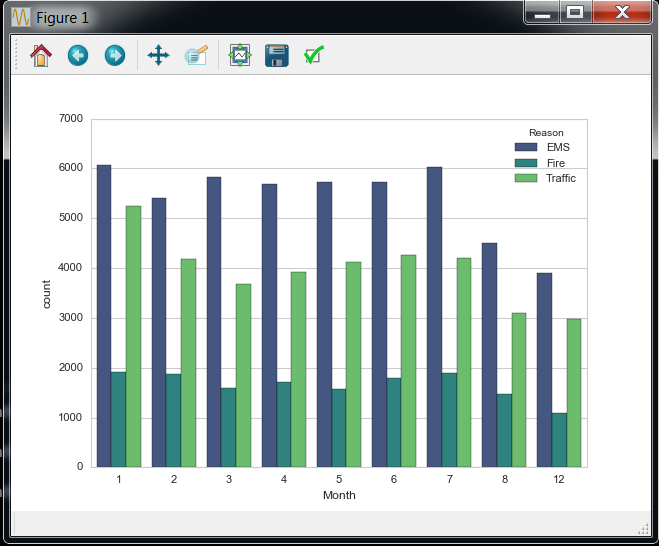
dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thu',4:'Fri',5:'Sat',6:'Sun'}

df['Day of Week'] = df['Day of Week'].map(dmap)

sns.countplot(x='Day of Week',data=df,hue='Reason',palette='viridis')

sns.plt.show()

**No. of calls based on different Reasons depending on Month:**



**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

df['timeStamp'] = pd.to\_datetime(df['timeStamp'])

df['Hour'] = df['timeStamp'].apply(lambda time: time.hour)

df['Month'] = df['timeStamp'].apply(lambda time: time.month)

df['Day of Week'] = df['timeStamp'].apply(lambda time: time.dayofweek)

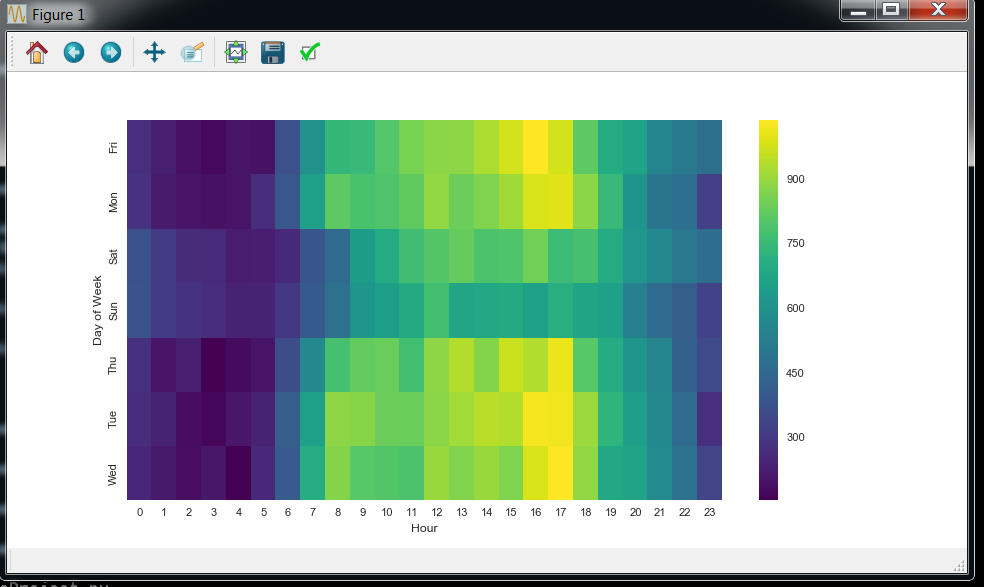
dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thu',4:'Fri',5:'Sat',6:'Sun'}

df['Day of Week'] = df['Day of Week'].map(dmap)

sns.countplot(x='Month',data=df,hue='Reason',palette='viridis')

sns.plt.show()

**Creating a Heatmap based on hour of the day:**

****

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

df['timeStamp'] = pd.to\_datetime(df['timeStamp'])

df['Hour'] = df['timeStamp'].apply(lambda time: time.hour)

df['Month'] = df['timeStamp'].apply(lambda time: time.month)

df['Day of Week'] = df['timeStamp'].apply(lambda time: time.dayofweek)

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)

dayHour = df.groupby(by=['Day of Week','Hour']).count()['Reason'].unstack()

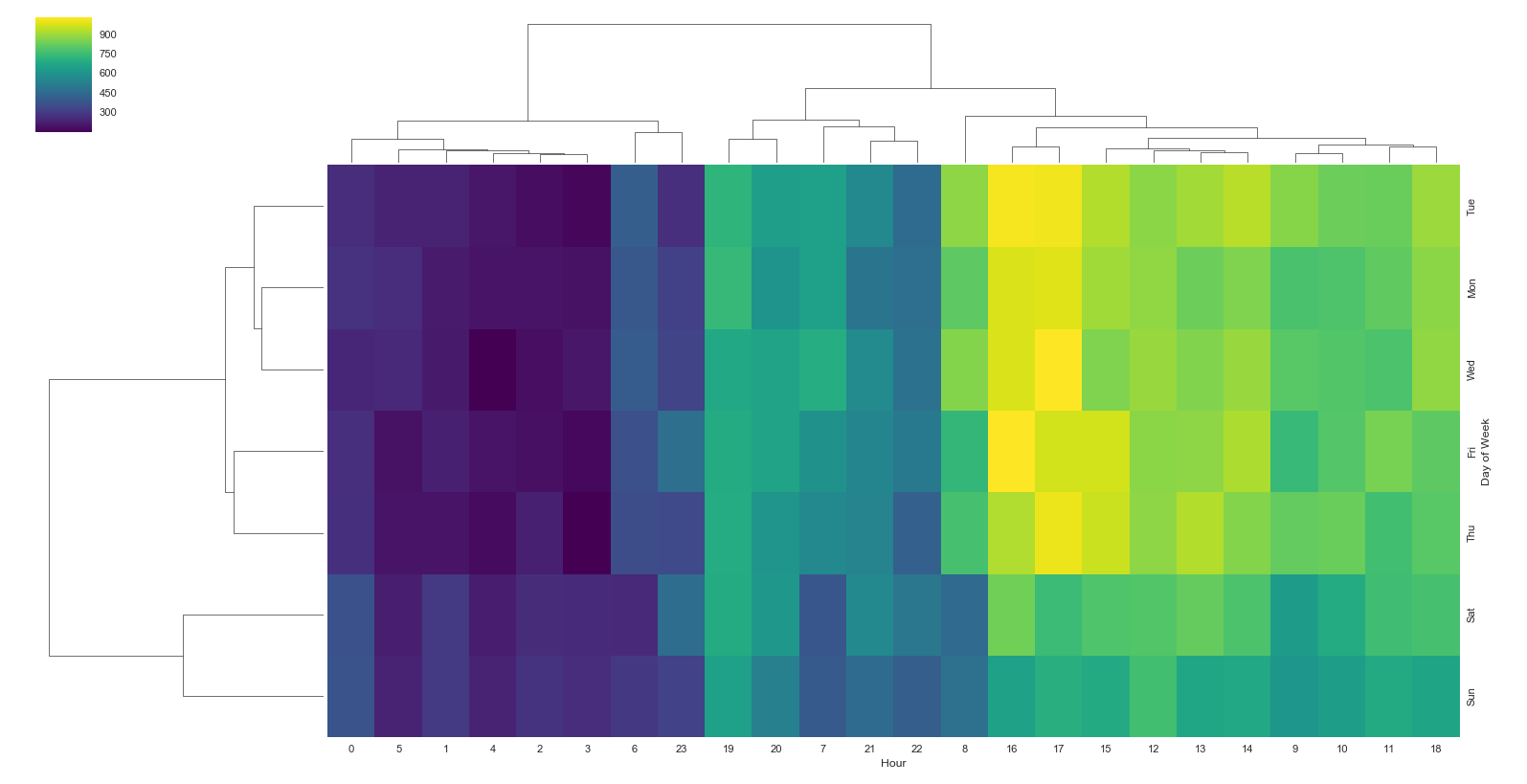
dayHour.head()

plt.figure(figsize=(12,6))

sns.heatmap(dayHour,cmap='viridis')

sns.plt.show()

**Creating a Cluster based on the time of the day:**



**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

df['timeStamp'] = pd.to\_datetime(df['timeStamp'])

df['Hour'] = df['timeStamp'].apply(lambda time: time.hour)

df['Month'] = df['timeStamp'].apply(lambda time: time.month)

df['Day of Week'] = df['timeStamp'].apply(lambda time: time.dayofweek)

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)

dayHour = df.groupby(by=['Day of Week','Hour']).count()['Reason'].unstack()

dayHour.head()

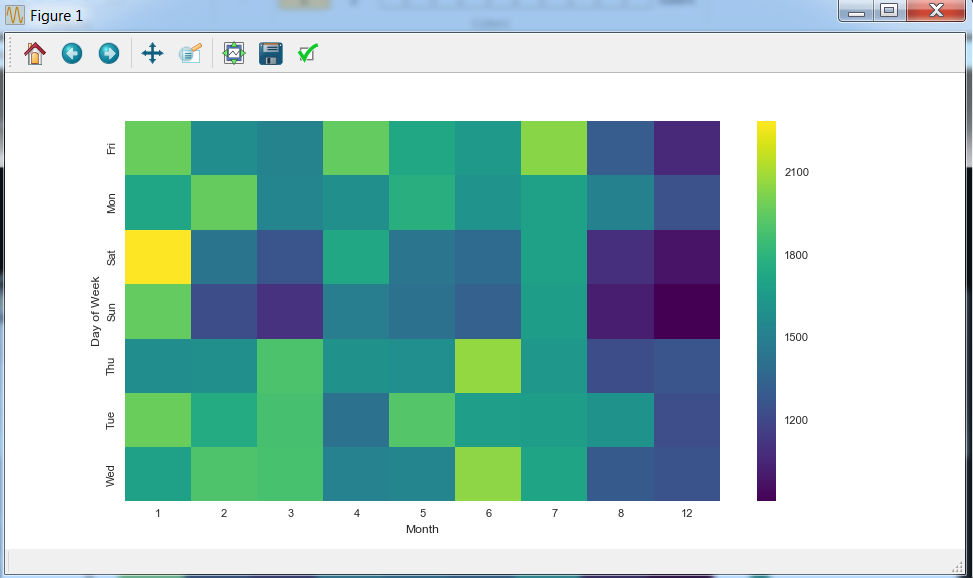
plt.figure(figsize=(12,6))

sns.heatmap(dayHour,cmap='viridis')

sns.clustermap(dayHour,cmap='viridis')

sns.plt.show()

**Heat Map Based on Month:**



**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

df['timeStamp'] = pd.to\_datetime(df['timeStamp'])

df['Hour'] = df['timeStamp'].apply(lambda time: time.hour)

df['Month'] = df['timeStamp'].apply(lambda time: time.month)

df['Day of Week'] = df['timeStamp'].apply(lambda time: time.dayofweek)

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)

dayMonth = df.groupby(by=['Day of Week','Month']).count()['Reason'].unstack()

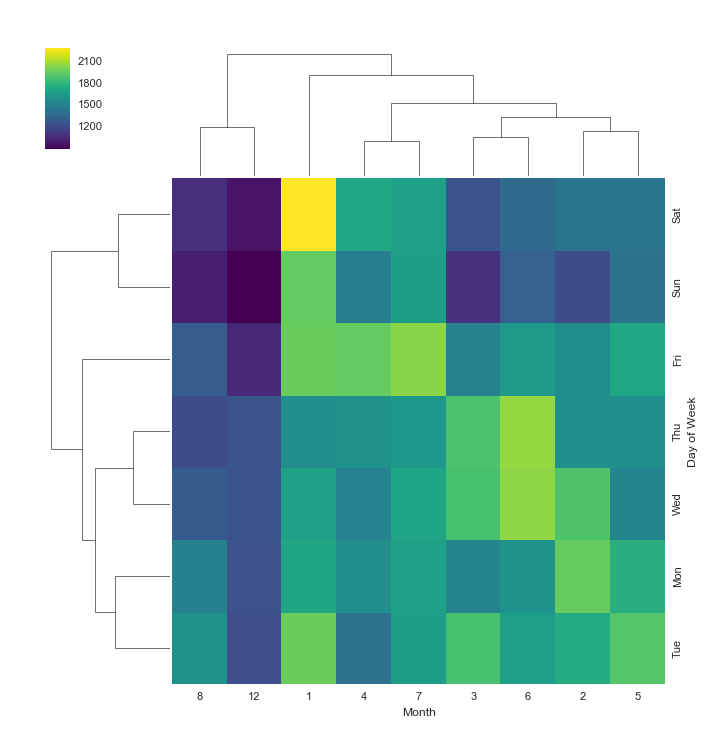
dayMonth.head()

plt.figure(figsize=(12,6))

sns.heatmap(dayMonth,cmap='viridis')

sns.plt.show()

ClusterMap Based on Month:



**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

df = pd.read\_csv('911.csv')

df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])

df['timeStamp'] = pd.to\_datetime(df['timeStamp'])

df['Hour'] = df['timeStamp'].apply(lambda time: time.hour)

df['Month'] = df['timeStamp'].apply(lambda time: time.month)

df['Day of Week'] = df['timeStamp'].apply(lambda time: time.dayofweek)

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)

dayMonth = df.groupby(by=['Day of Week','Month']).count()['Reason'].unstack()

dayMonth.head()

sns.clustermap(dayMonth,cmap='viridis')

sns.plt.show()