Initializing System

```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
```

```
!tar xf "/content/drive/My Drive/BigDataAssignment3Files/spark-2.4.5-bin-hadoop2.7.tgz"
!pip install -q findspark
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-2.4.5-bin-hadoop2.7"
```

```
import findspark
findspark.init()
from pyspark.sql import SparkSession
from pyspark.context import SparkContext
spark = SparkSession.builder.master("local[*]").getOrCreate()
```

Reading Preprocessed Dataset

```
preprocessed_data = spark.read.json("hdfs://udit_gupta_1/processed_data")
```

Extract Data for EDA

```
#Get top 30 crime types in complaints
top30_crime_type = preprocessed_data.rdd \
.filter(lambda row : row['RECORD_TYPE'] == 'C') \
.map(lambda row : (row['OFNS_DESC'],1)) \
.reduceByKey(lambda key1, key2 : key1 + key2) \
.takeOrdered(30,lambda atuple: -atuple[1])
```

```
complaints_crime_list = [ele[0] for ele in top30_crime_type if ele[0] is not None]
```

```
#Get top 30 arrests crime types
top30_arrests_crime_type = preprocessed_data.rdd \
.filter(lambda row : row['RECORD_TYPE'] == 'A') \
.map(lambda row : (row['OFNS_DESC'],1)) \
.reduceByKey(lambda key1, key2 : key1 + key2) \
.takeOrdered(30,lambda atuple: -atuple[1])
```

```
arrests_crime_list = [ele[0] for ele in top30_arrests_crime_type if ele[0] is not None]
```

```
#Get top 30 location types for crime complaints
top30_crime_locations = preprocessed_data.rdd \
.filter(lambda row : row['RECORD_TYPE'] == 'C') \
.map(lambda row : (row['PREM_TYP_DESC'],1)) \
.reduceByKey(lambda key1, key2 : key1 + key2) \
.takeOrdered(30,lambda atuple: -atuple[1])
```

```
complaints_location_list = [ele[0] for ele in top30_crime_locations if ele[0] is not None]
```

Generic Imports

```
import pandas as pd
import plotly.express as px
from pyspark.sql.functions import unix_timestamp, from_unixtime
from pyspark.sql import functions as F
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import folium
from folium.plugins import HeatMap

%matplotlib inline
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:

pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
```

Time Series Analysis

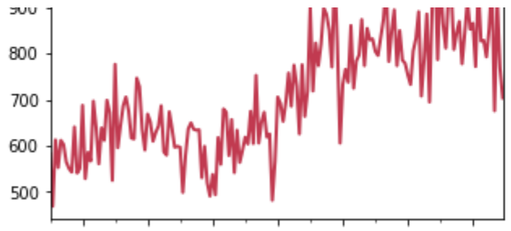
```
arrests_df_casted = preprocessed_data.filter(preprocessed_data['RECORD_TYPE'] == 'A').select("*",from_unixtime(unix_timestamp(
```

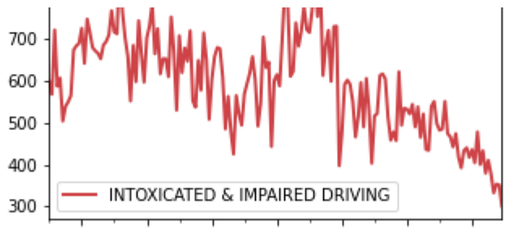
Small Multiples for Major Crimes Types from 2005 to 2018 on monthly grouping

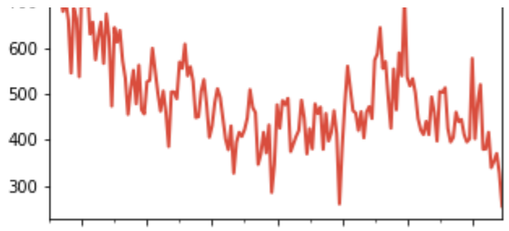
```
` vs Number of Crime for Each Crime Type
ımmarized data to pandas for easy plotting
ṛpe_pivot = arrests_df_casted.groupby('CastedDate').pivot("OFNS_DESC", arrests_crime_list).agg({"*": "count"}).fillna(0).toPanc
ṛpe_pivot['CastedDate'] = pd.to_datetime(day_vs_crimetype_pivot['CastedDate'])
ṛpe_pivot = day_vs_crimetype_pivot.set_index('CastedDate')
```

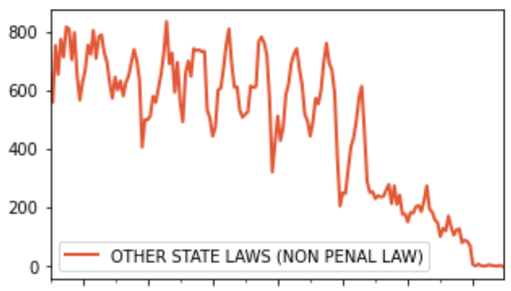
```
#Generate Plot
day_vs_crimetype_pivot.resample('M').sum().plot(figsize=(17,40), linewidth=2, cmap='inferno', subplots=True, layout=(-1, 3))
plt.legend(prop={'size':5})
plt.tick_params(labelsize=8)
```

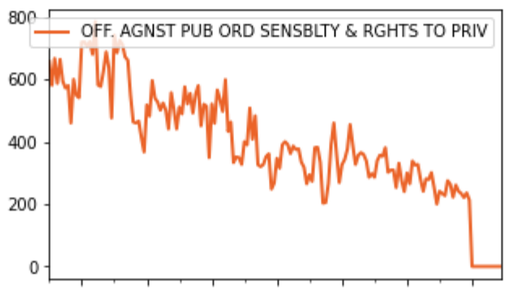
DANGEROUS DRUGS
ASSAULT 3 & RELATED OFFENSES
OTHER OFFENSES RELATED TO THEFT
OTHER STATE LAWS
PETIT LARCENY
VEHICLE AND TRAFFIC LAWS
CRIMINAL TRESPASS
FELONY ASSAULT
DANGEROUS WEAPONS
MISCELLANEOUS PENAL LAW
OTHER TRAFFIC INFRACTION
OFFENSES AGAINST PUBLIC ADMINISTRATION
ROBBERY
POSSESSION OF STOLEN PROPERTY 5
CRIMINAL MISCHIEF & RELATED OFFENSES
GRAND LARCENY
FORGERY

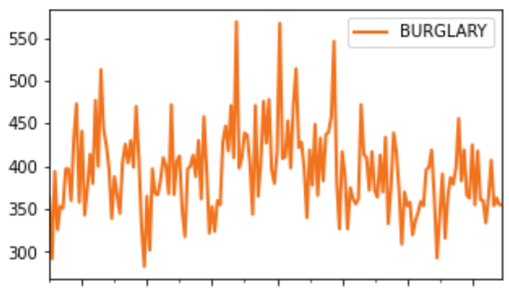


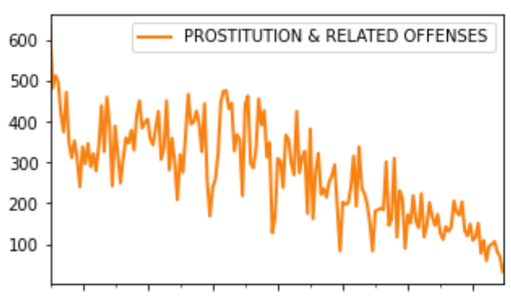


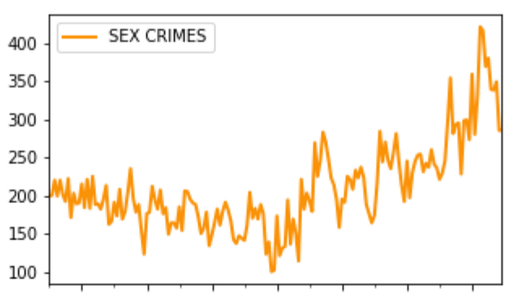


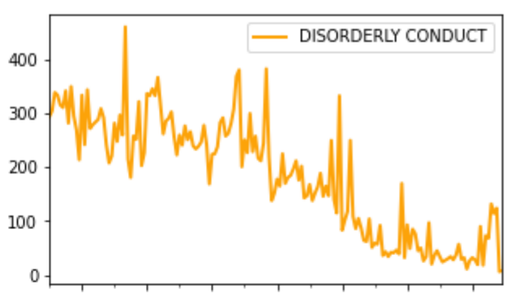


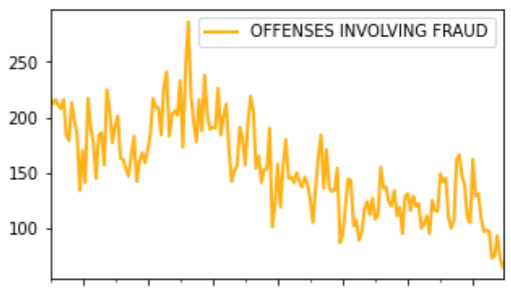


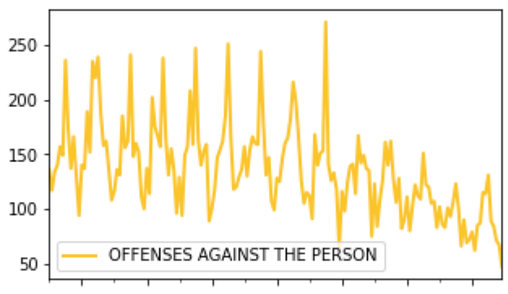


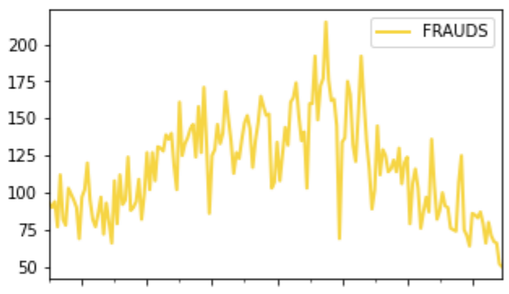










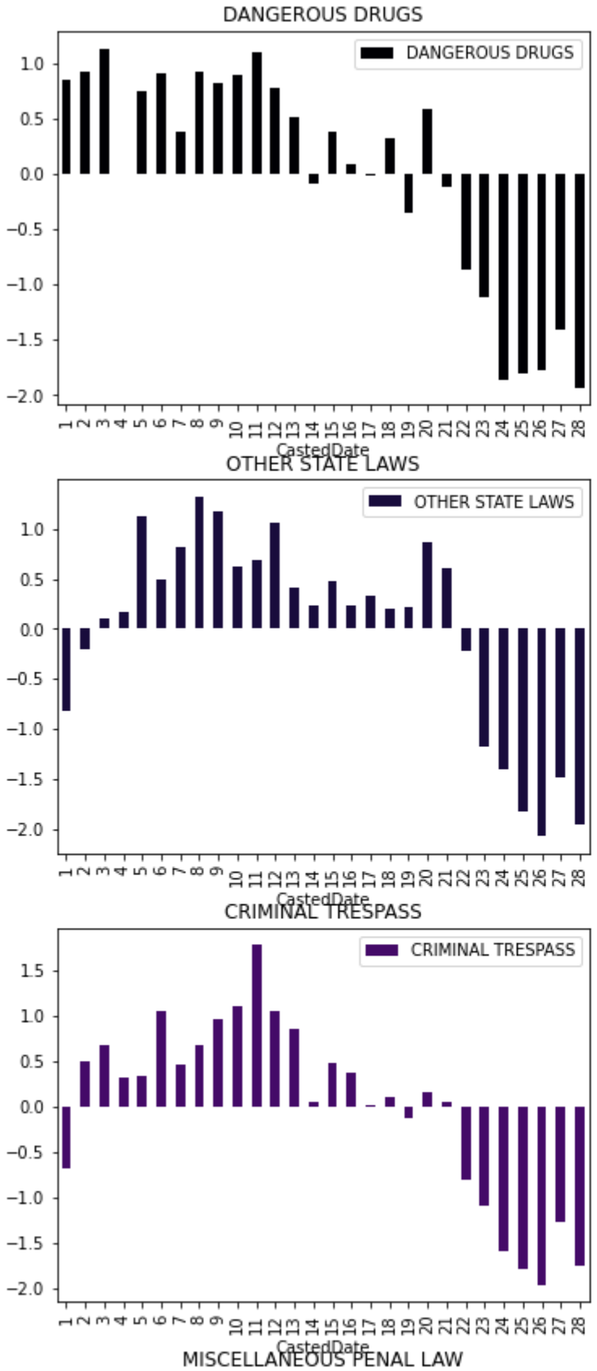


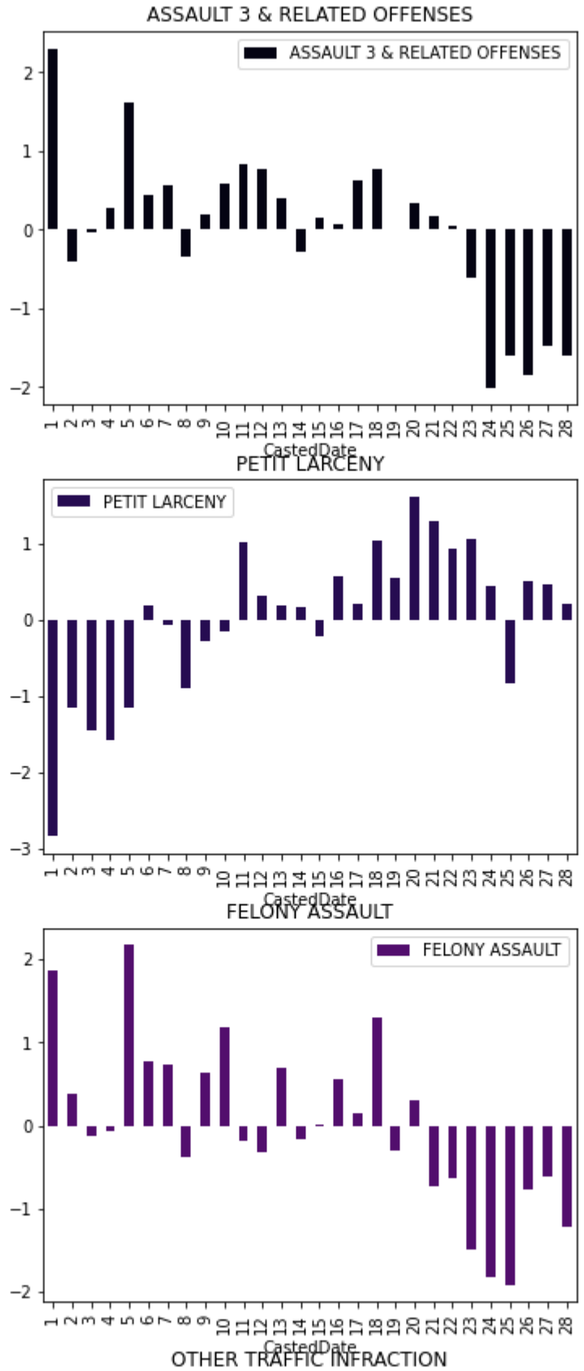

```
#Helper Functions
from sklearn.cluster import AgglomerativeClustering as AC

def scale_df(df,axis=0):
    '''
    A utility function to scale numerical values (z-scale) to have a mean of zero
    and a unit variance.
    '''
    return (df - df.mean(axis=axis)) / df.std(axis=axis)

def plot_hmap(df, ix=None, cmap='bwr'):
    '''
    A function to plot heatmaps that show temporal patterns
    '''
```

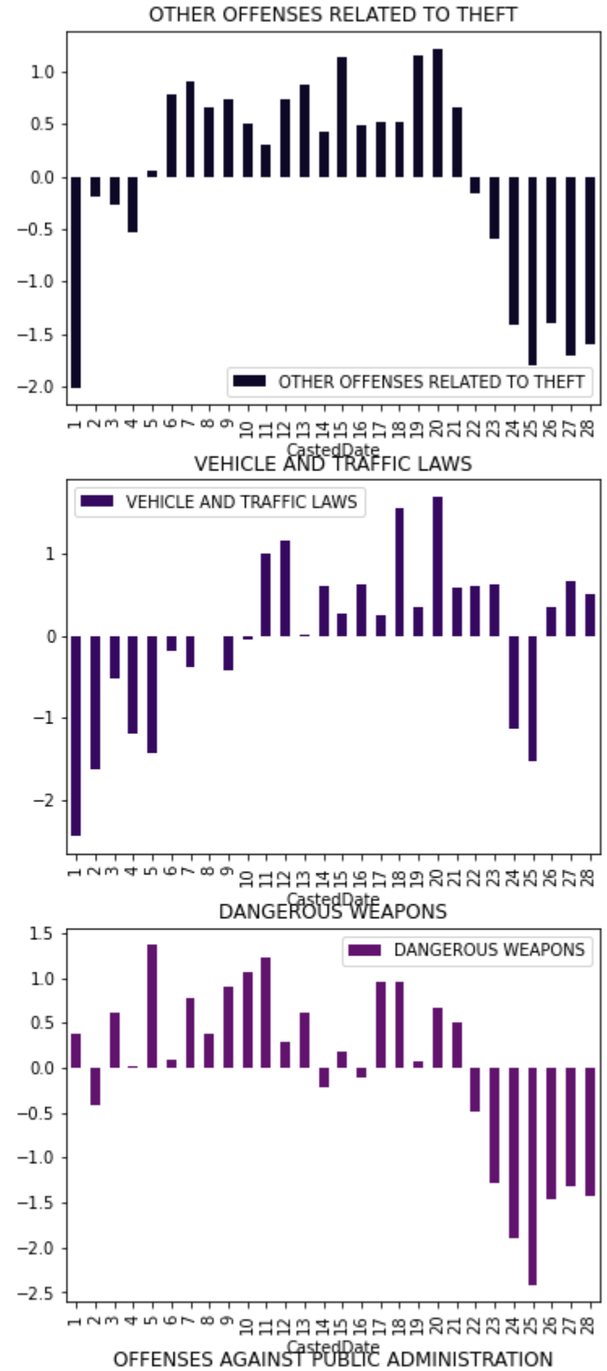
```
    if ix is None:
        ix = np.arange(df.shape[0])
    plt.imshow(df.iloc[ix,:], cmap=cmap)
    plt.colorbar(fraction=0.03)
    plt.yticks(np.arange(df.shape[0]), df.index[ix])
    plt.xticks(np.arange(df.shape[1]))
    plt.grid(False)
    plt.show()

#Plots hmap for scaled data with an additional clustering step for better visuals.
def scale_and_plot(df, ix = None):
    '''
    A wrapper function to calculate the scaled values within each row of df and plot_hmap
    '''
    df_marginal_scaled = scale_df(df.T).T
    if ix is None:
        ix = AC(4).fit(df_marginal_scaled).labels_.argsort()
    cap = np.min([np.max(df_marginal_scaled.as_matrix()), np.abs(np.min(df_marginal_scaled.as_matrix()))])
    df_marginal_scaled = np.clip(df_marginal_scaled, -1*cap, cap)
    plot_hmap(df_marginal_scaled, ix=ix)
    return df_marginal_scaled

#Normalizes Data with min-max norm
def normalize(df):
    result = df.copy()
    for feature_name in df.columns:
        max_value = df[feature_name].max()
        min_value = df[feature_name].min()
        result[feature_name] = (df[feature_name] - min_value) / (max_value - min_value)
    return result

```

Small Multiples of Crime Types vs Day of Month

```
#Plotting only for first 28 days since 29th, 30th and 31st is not present in Feb
scale_df(day_vs_crimetype_pivot.groupby(day_vs_crimetype_pivot.index.day).sum().iloc[:28]).plot(kind='bar', figsize=(20,50), l
plt.legend(prop={'size':5})
plt.xlabel('')
plt.tick_params(labelsize=5)
```


DANGEROUS DRUGS
DANGEROUS DRUGS
CastedDate


ASSAULT 3 & RELATED OFFENSES
ASSAULT 3 & RELATED OFFENSES
CastedDate


OTHER OFFENSES RELATED TO THEFT
OTHER OFFENSES RELATED TO THEFT
CastedDate

OTHER STATE LAWS
OTHER STATE LAWS
CastedDate

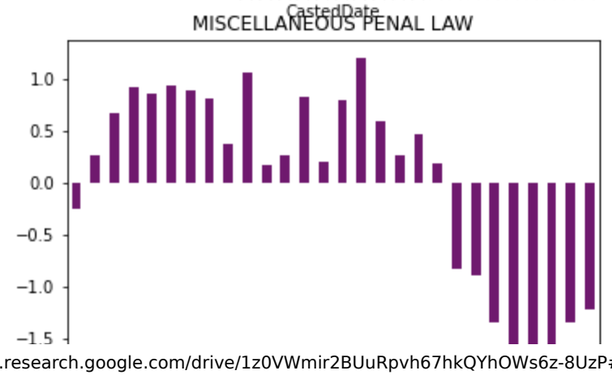
PETIT LARCENY
PETIT LARCENY
CastedDate

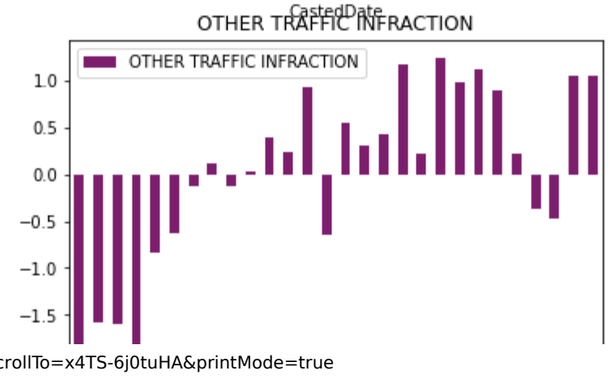
VEHICLE AND TRAFFIC LAWS
VEHICLE AND TRAFFIC LAWS
CastedDate

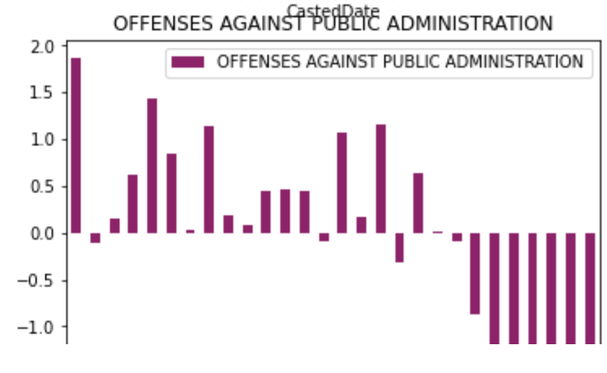
CRIMINAL TRESPASS
CRIMINAL TRESPASS
CastedDate

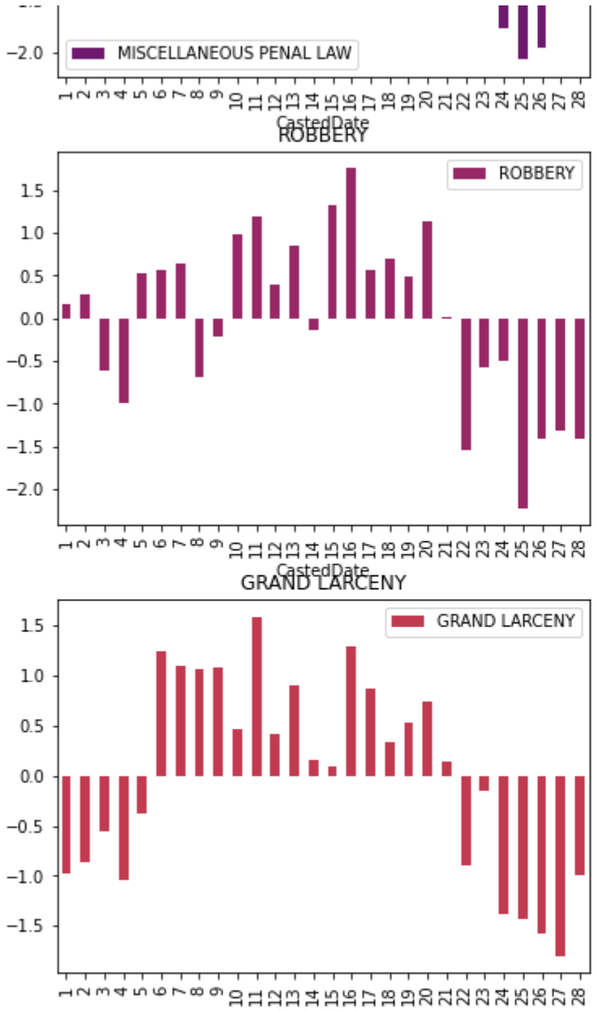
FELONY ASSAULT
FELONY ASSAULT
CastedDate

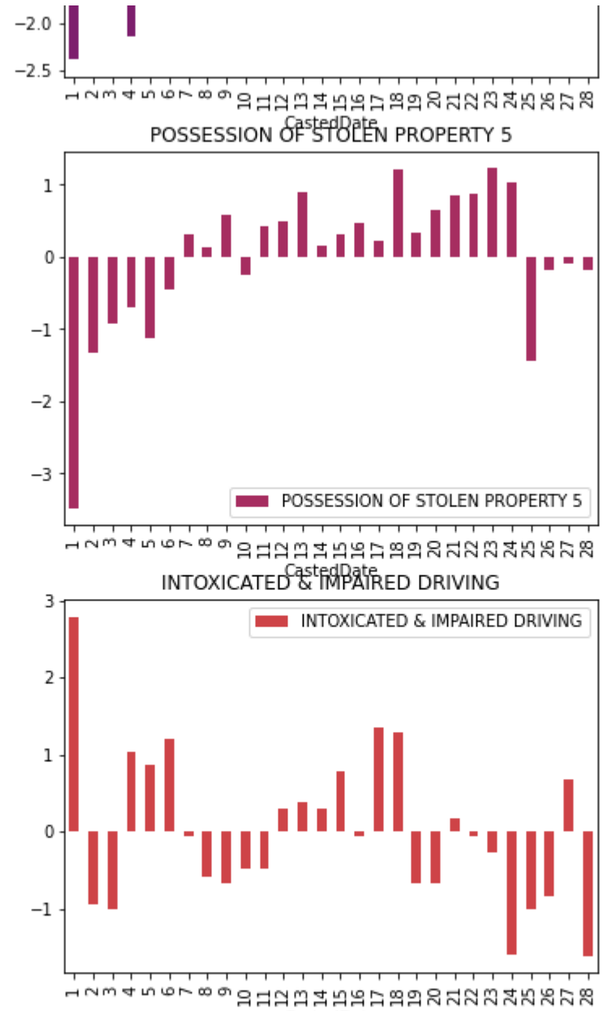
DANGEROUS WEAPONS
DANGEROUS WEAPONS
CastedDate

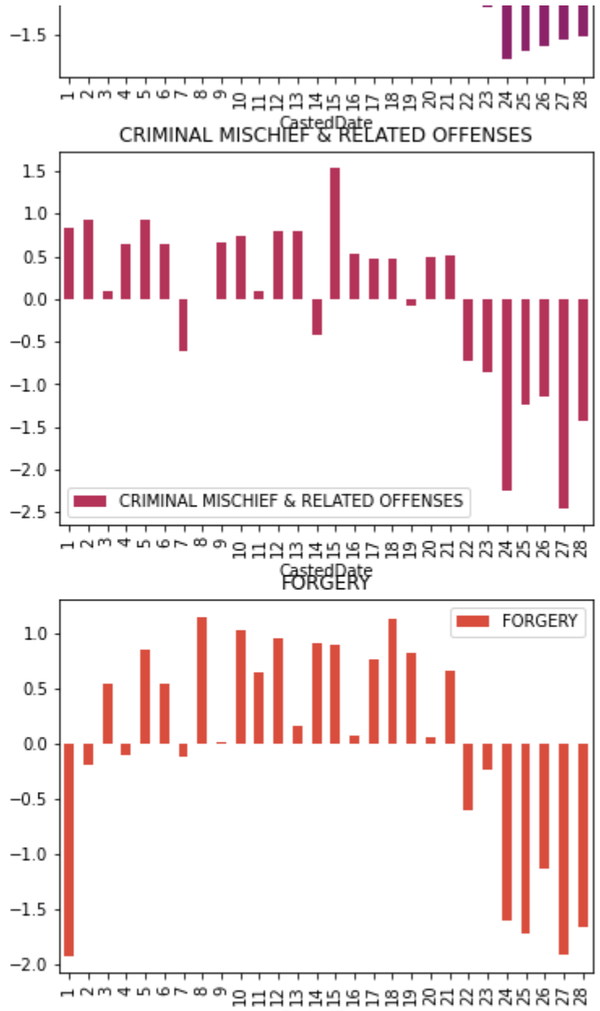

MISCELLANEOUS PENAL LAW


OTHER TRAFFIC INFRACTION
OTHER TRAFFIC INFRACTION


OFFENSES AGAINST PUBLIC ADMINISTRATION
OFFENSES AGAINST PUBLIC ADMINISTRATION


MISCELLANEOUS PENAL LAW
CastedDate


CastedDate


CastedDate

ROBBERY
ROBBERY
CastedDate

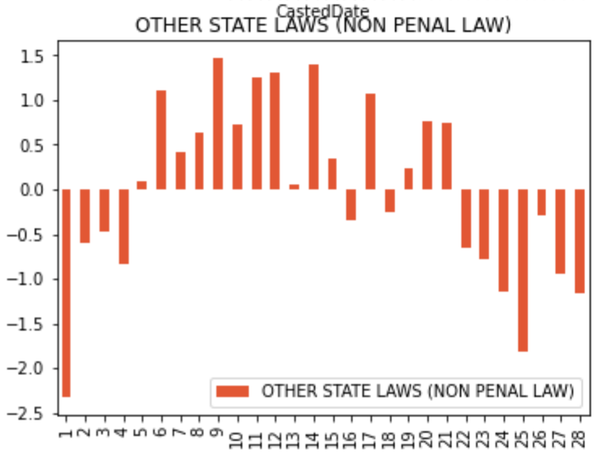
POSSESSION OF STOLEN PROPERTY 5
POSSESSION OF STOLEN PROPERTY 5
CastedDate

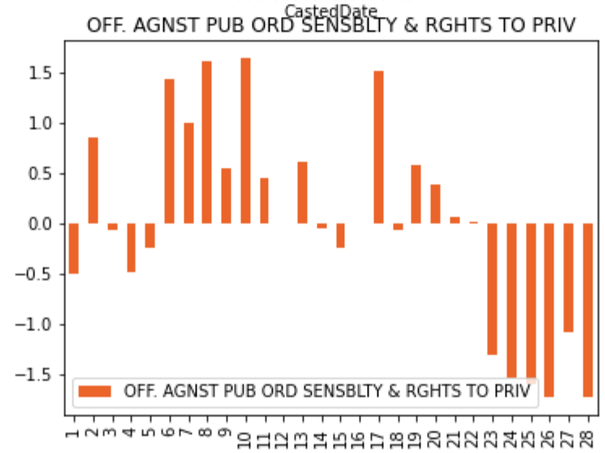
CRIMINAL MISCHIEF & RELATED OFFENSES
CRIMINAL MISCHIEF & RELATED OFFENSES
CastedDate

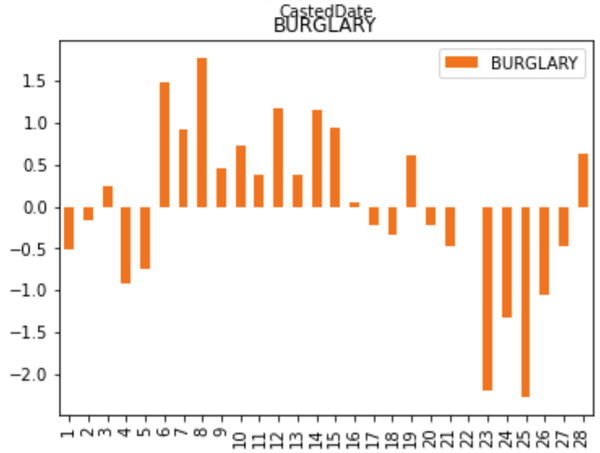
GRAND LARCENY
GRAND LARCENY
CastedDate

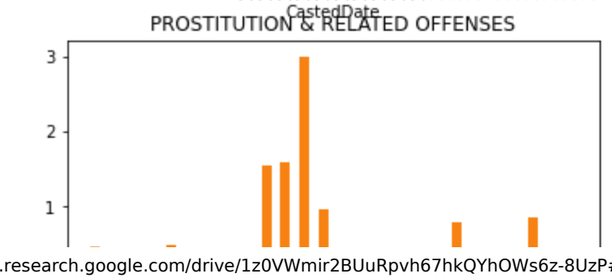
INTOXICATED & IMPAIRED DRIVING
INTOXICATED & IMPAIRED DRIVING
CastedDate

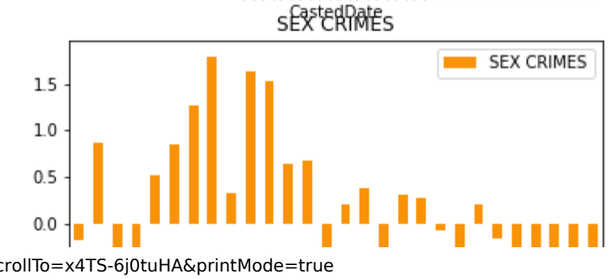
FORGERY
FORGERY
CastedDate

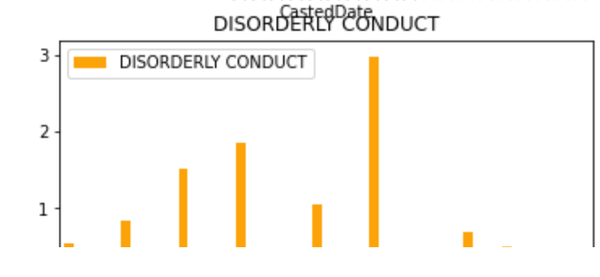

OTHER STATE LAWS (NON PENAL LAW)
OTHER STATE LAWS (NON PENAL LAW)
CastedDate


OFF. AGNST PUB ORD SENSBLTY & RGHTS TO PRIV
OFF. AGNST PUB ORD SENSBLTY & RGHTS TO PRIV
CastedDate


BURGLARY
BURGLARY
CastedDate

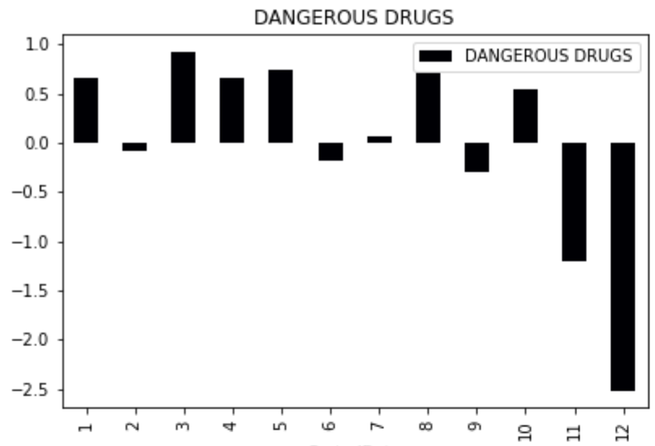

PROSTITUTION & RELATED OFFENSES

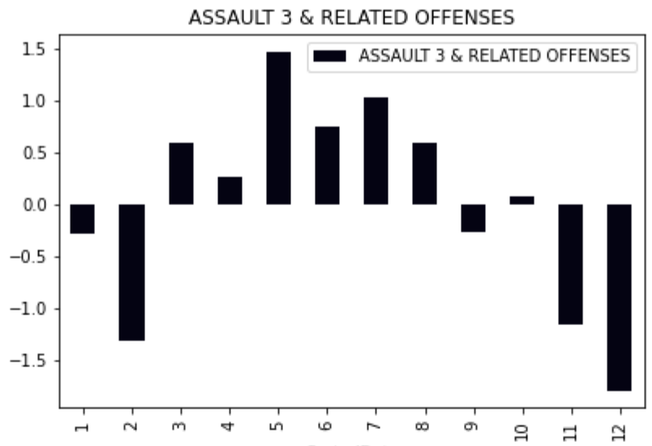

SEX CRIMES
SEX CRIMES

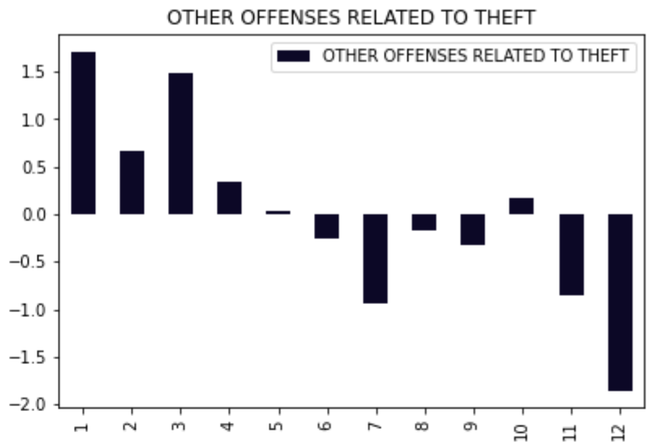

DISORDERLY CONDUCT
DISORDERLY CONDUCT

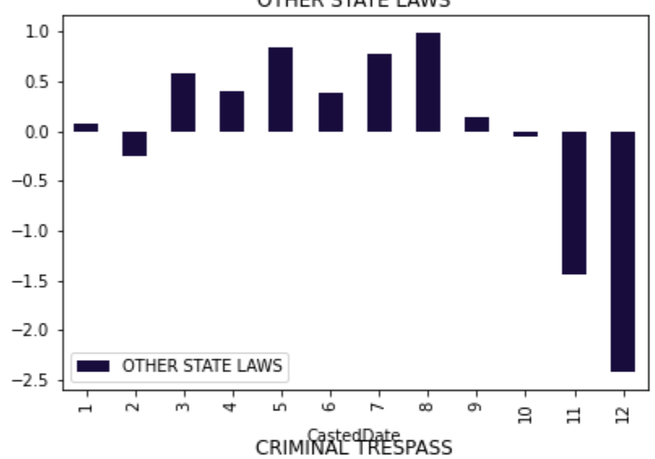
Small Multiples of Crime Types vs Month of Year

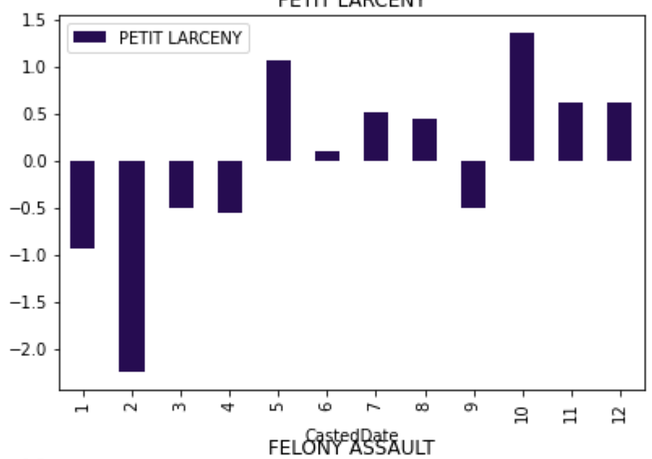
```
cale_df(day_vs_crimetype_pivot.groupby(day_vs_crimetype_pivot.index.month).sum()) \
plot(kind='bar', figsize=(22,50), linewidth=2, cmap='inferno', subplots=True, layout=(-1, 3), sharex=False, sharey=False)
lt.show()
```

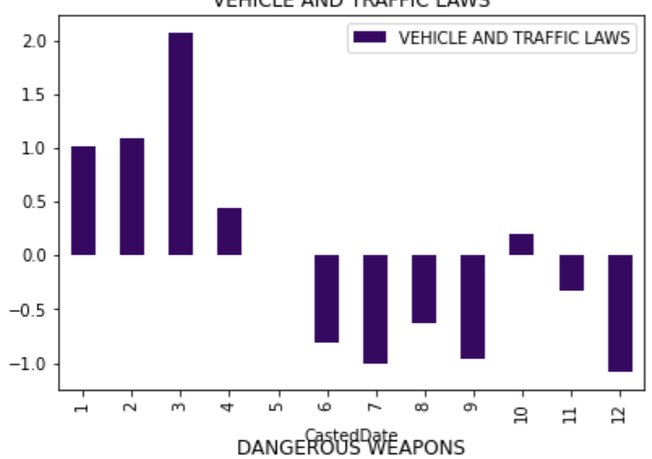

DANGEROUS DRUGS
DANGEROUS DRUGS
CastedDate

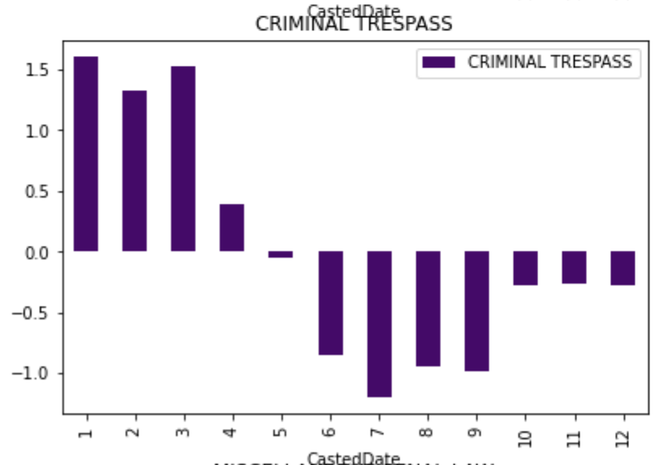

ASSAULT 3 & RELATED OFFENSES
ASSAULT 3 & RELATED OFFENSES
CastedDate

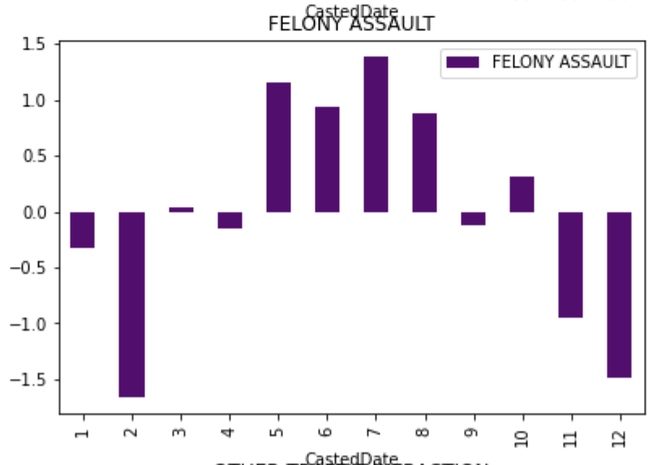

OTHER OFFENSES RELATED TO THEFT
OTHER OFFENSES RELATED TO THEFT
CastedDate

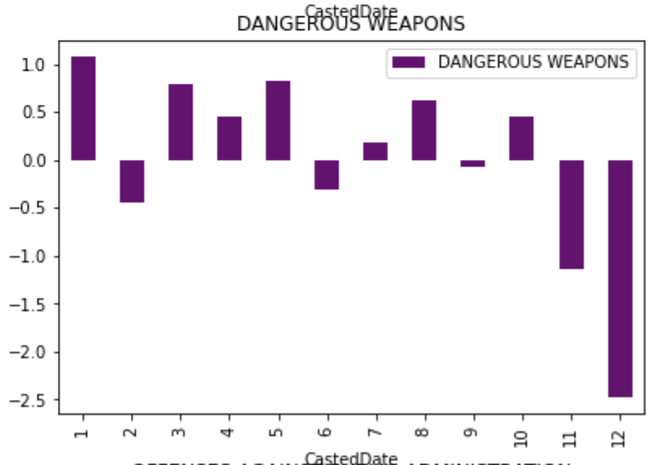

OTHER STATE LAWS
OTHER STATE LAWS
CastedDate

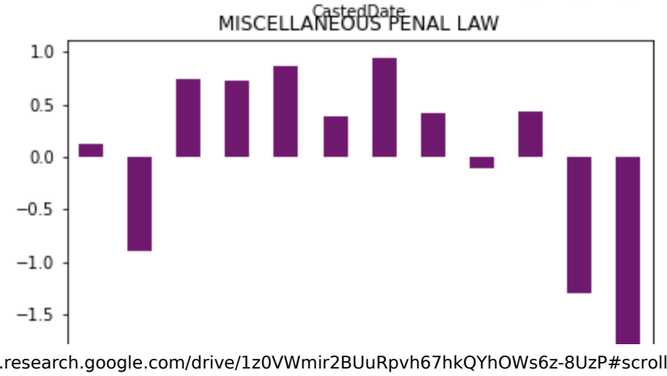

PETIT LARCENY
PETIT LARCENY
CastedDate

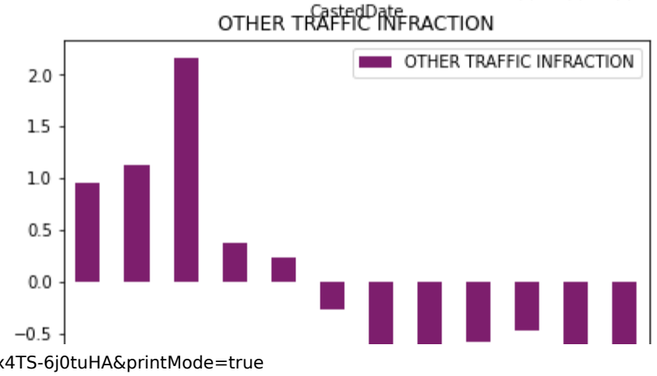

VEHICLE AND TRAFFIC LAWS
VEHICLE AND TRAFFIC LAWS
CastedDate

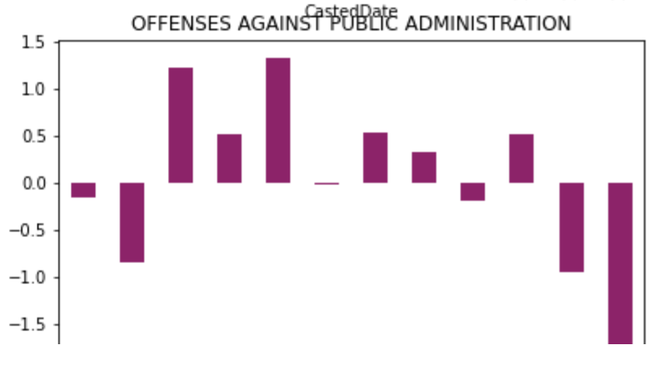

CRIMINAL TRESPASS
CRIMINAL TRESPASS
CastedDate


FELONY ASSAULT
FELONY ASSAULT
CastedDate


DANGEROUS WEAPONS
DANGEROUS WEAPONS
CastedDate


MISCELLANEOUS PENAL LAW


OTHER TRAFFIC INFRACTION
OTHER TRAFFIC INFRACTION


OFFENSES AGAINST PUBLIC ADMINISTRATION