United States CRIME ANALYSIS

Data Set Description:

There were multiple data sets used in this crime analysis. Individual crime data sets for the cities of Portland, Houston, Los Angeles, Atlanta, Seattle, Boston, and Chicago were collected from Kaggle for the years 2012 - 2017. Additionaly, data sets containing weather data and weather descriptions were also obtained from Kaggle. Finally, a data set containing populations for the previously mentioned cities was used to calculate crime rates. All of this data was merged in order to find connections between weather and crime as well as day of the week and crime.

All the above datasets are merged together to result in a final dataset that contains:

Columns :

Datetime : Date and time of incident in hourly intervals

Offense_Num: Types of crime(1 = Theft, 2 = Burglary, 3 = assault , 4 = sex crimes , 5 = forgery, 6 = weapons , 7 = drug , 8 = fraud, 9 = arson , 10 = Kidnapping , 11 = Homicide , 12 = Harassment, 13 = Driving, 14 = Other

Offense_Type: Deatailed description of the crime

Temp: Temperatue in degrees Farenheit

Bins : The temperatures are classified into 10 degree bins in degrees Farenheit

TD_num: Types of weather conditions (1 = Thunderstorn , 2 = Drizzle, 3 = Rain, 4 = Snow , 5 = Atmosphere, 6 = Clear, 7 = Clouds)

Temp_des: Detailed description of the weather condition

Year: Year when crime occured

city: Name of the city where crime occured

Rows :

3,673,957 total rows

Data Preparation:

```
In [1]: # print all the outputs in a cell
        from IPython.core.interactiveshell import InteractiveShell
        InteractiveShell.ast_node_interactivity = "all"
        import seaborn as sns
        import pandas as pd
        import numpy as np
        %pylab inline
```

```
        Populating the interactive namespace from numpy and matplotlib
```

```
In [2]: import warnings
        warnings.filterwarnings("ignore")
```

```
In [3]: %autosave 0
```

```
        Autosave disabled
```

CLEANING OF WEATHER DATA

Loading temperature data

```
In [4]: df_temp = pd.read_csv('temperature.csv',usecols=['datetime', 'Portland
        ', 'Seattle', \
                                                                 'Los Angeles','Housto
        n','Chicago','Atlanta','Boston'])
```

```
In [5]: df_temp.head(2)
```

Out[5]:

	datetime	Portland	Seattle	Los Angeles	Houston	Chicago	Atlanta	Boston
0	2012-10-01 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2012-10-01 13:00:00	282.08	281.8	291.87	288.27	284.01	294.03	287.17

Creating copy of temperature data

```
In [6]: df_temp_copy=df_temp.copy()
```

Dropping all Null Value Columns

```
In [7]: df_temp_copy = df_temp_copy.dropna(how='any',axis=0)
```

```
In [8]: df_temp_copy.isna().any()
```

```
Out[8]: datetime       False
        Portland       False
        Seattle        False
        Los Angeles    False
        Houston        False
        Chicago        False
        Atlanta        False
        Boston         False
        dtype: bool
```

Changing Temperature into Farenheit

```
In [9]: df_temp_copy.Portland= 1.8 * (df_temp_copy['Portland'] - 273.15) +32
```

```
In [10]: df_temp_copy['Los Angeles']= 1.8 * (df_temp_copy['Los Angeles'] - 273.
         15) +32
```

```
In [11]: df_temp_copy.Houston= 1.8 * (df_temp_copy['Houston'] - 273.15) +32
```

```
In [12]: df_temp_copy.Chicago= 1.8 * (df_temp_copy['Chicago'] - 273.15) +32
```

```
In [13]: df_temp_copy.Atlanta= 1.8 * (df_temp_copy['Atlanta'] - 273.15) +32
```

```
In [14]: df_temp_copy.Boston= 1.8 * (df_temp_copy['Boston'] - 273.15) +32
```

```
In [15]: df_temp_copy.Seattle= 1.8 * (df_temp_copy['Seattle'] - 273.15) +32
```

In [16]:
```
df_temp_copy.head(2)
```

Out[16]:

	datetime	Portland	Seattle	Los Angeles	Houston	Chicago	Atlanta	Bost
1	2012-10-01 13:00:00	48.074000	47.57000	65.696000	59.216000	51.548000	69.584000	57.2360
2	2012-10-01 14:00:00	48.079854	47.56499	65.692734	59.265636	51.628444	69.593615	57.2649

Creating Temperature bins

In [17]:
```
temp_bins = pd.IntervalIndex.from_tuples([(-20, -10),(-10, 0),(0, 10),
(10, 20),(20, 30),\
                                          (30, 40),(40, 50),(50, 60),(
60,70),(70,80),\
                                          (90,100),(100,120)])
```

Adding temperature bins for all the cities

In [18]:
```
df_temp_copy['Portland_bin']=pd.cut(x=df_temp_copy.Portland,bins=temp_
bins,right=True)
```

In [19]:
```
df_temp_copy['Los Angeles_bin']=pd.cut(x=df_temp_copy['Los Angeles'],b
ins=temp_bins,right=True)
```

In [20]:
```
df_temp_copy['Houston_bin']=pd.cut(x=df_temp_copy.Houston,bins=temp_bi
ns,right=True)
```

In [21]:
```
df_temp_copy['Chicago_bin']=pd.cut(x=df_temp_copy.Chicago,bins=temp_bi
ns,right=True)
```

In [22]:
```
df_temp_copy['Atlanta_bin']=pd.cut(x=df_temp_copy.Atlanta,bins=temp_bi
ns,right=True)
```

In [23]:
```
df_temp_copy['Boston_bin']=pd.cut(x=df_temp_copy.Boston,bins=temp_bins
,right=True)
```

In [24]:
```
df_temp_copy['Seattle_bin']=pd.cut(x=df_temp_copy.Seattle,bins=temp_bi
ns,right=True)
```

In [25]:
```
df_temp_copy.head(2)
```

Out[25]:

	datetime	Portland	Seattle	Los Angeles	Houston	Chicago	Atlanta	Bost
1	2012-10-01 13:00:00	48.074000	47.57000	65.696000	59.216000	51.548000	69.584000	57.2360
2	2012-10-01 14:00:00	48.079854	47.56499	65.692734	59.265636	51.628444	69.593615	57.2649

Loading Weather Description Data

In [26]:
```
df_weather = pd.read_csv('weather_description.csv',usecols=['datetime'
, 'Portland', 'Seattle', \
                                                   'Los Angeles','Housto
n','Chicago','Atlanta','Boston'])
```

Making a copy of weather description data

In [27]:
```
df_weather_c=df_weather.copy()
```

In [28]:
```
df_weather_c.head()
```

Out[28]:

	datetime	Portland	Seattle	Los Angeles	Houston	Chicago	Atlanta	Boston
0	2012-10-01 12:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2012-10-01 13:00:00	scattered clouds	sky is clear	mist	sky is clear	overcast clouds	light rain	sky is clear
2	2012-10-01 14:00:00	scattered clouds	sky is clear	sky is clear	sky is clear	overcast clouds	overcast clouds	few clouds
3	2012-10-01 15:00:00	scattered clouds	sky is clear	sky is clear	sky is clear	overcast clouds	overcast clouds	few clouds
4	2012-10-01 16:00:00	scattered clouds	sky is clear	sky is clear	sky is clear	overcast clouds	overcast clouds	few clouds

Dropping NA values from dataset

```
In [29]: df_weather_c = df_weather_c.dropna(how='any',axis=0)
```

```
In [30]: df_weather_c.head(2)
```

```
Out[30]:
```

	datetime	Portland	Seattle	Los Angeles	Houston	Chicago	Atlanta	Boston
1	2012-10-01 13:00:00	scattered clouds	sky is clear	mist	sky is clear	overcast clouds	light rain	sky is clear
2	2012-10-01 14:00:00	scattered clouds	sky is clear	sky is clear	sky is clear	overcast clouds	overcast clouds	few clouds

```
In [31]: df_weather_c.isna().any()
```

```
Out[31]: datetime       False
         Portland       False
         Seattle        False
         Los Angeles    False
         Houston        False
         Chicago        False
         Atlanta        False
         Boston         False
         dtype: bool
```

```
In [32]: df_weather_c.head(2)
```

```
Out[32]:
```

	datetime	Portland	Seattle	Los Angeles	Houston	Chicago	Atlanta	Boston
1	2012-10-01 13:00:00	scattered clouds	sky is clear	mist	sky is clear	overcast clouds	light rain	sky is clear
2	2012-10-01 14:00:00	scattered clouds	sky is clear	sky is clear	sky is clear	overcast clouds	overcast clouds	few clouds

Categorizing the types of weather into 7 different categories

1.) Thunderstorn

2.) Drizzle

3.) Rain

4.) Snow

5.) Atmosphere

6.) Clear

7.) Clouds

In [33]:

```
def Climate_des(JobString):
    if JobString in ('thunderstorm','thunderstorm with light rain', 't
hunderstorm with heavy rain',\
        'thunderstorm with rain','light thunderstorm','thunderstorm','
heavy thunderstorm',\
        'ragged thunderstorm','thunderstorm with light drizzle','thund
erstorm with drizzle',\
       'thunderstorm with heavy drizzle','proximity thunderstorm'):
        return 1
    elif JobString in ('light intensity drizzle','drizzle','high inten
sity drizzle',\
                      'light intensity drizzle','drizzle rain','heavy
intensity drizzle',\
                      'shower rain and drizzle','heavy shower rain and
drizzle',\
                      'proximity thunderstorm with drizzle','shower dr
izzle'):
        return 2
    elif JobString in ('light rain','moderate rain','heavy intensity r
ain','very heavy rain',\
                      'extreme rain','freezing rain','light intensity
shower rain','shower rain',\
                      'heavy intensity shower rain','proximity shower
rain','proximity moderate rain',\
                       'proximity thunderstorm with rain','ragged show
er rain'):
        return 3
    elif JobString in ('light snow','heavy snow','snow','sleet','showe
r sleet','light rain and snow',\
                      'rain and snow','light shower snow','shower snow
','heavy shower snow'):
        return 4
    elif JobString in ('mist','smoke','haze','sand','dust','fog','sand
,dust whirls','volcanic ash',\
                      'squalls','tornado'):
        return 5
    elif JobString in ('sky is clear','Portland_grouping'):
        return 6
    elif JobString in ('overcast clouds','broken clouds','scattered cl
ouds','few clouds'):
        return 7
```

In [34]:
```
df_weather_c['Portland_grouping'] = df_weather_c['Portland'].apply(Cli
mate_des)
df_weather_c['Seattle_grouping'] = df_weather_c['Seattle'].apply(Clima
te_des)
df_weather_c['LosAngeles_grouping'] = df_weather_c['Los Angeles'].appl
y(Climate_des)
df_weather_c['Houston_grouping'] = df_weather_c['Houston'].apply(Clima
te_des)
df_weather_c['Chicago_grouping'] = df_weather_c['Chicago'].apply(Clima
te_des)
df_weather_c['Atlanta_grouping'] = df_weather_c['Atlanta'].apply(Clima
te_des)
df_weather_c['Boston_grouping'] = df_weather_c['Boston'].apply(Climate
_des)
```

In [35]:
```
df_weather_c.head(2)
```

Out[35]:

	datetime	Portland	Seattle	Los Angeles	Houston	Chicago	Atlanta	Boston	Portland_
1	2012-10-01 13:00:00	scattered clouds	sky is clear	mist	sky is clear	overcast clouds	light rain	sky is clear	7
2	2012-10-01 14:00:00	scattered clouds	sky is clear	sky is clear	sky is clear	overcast clouds	overcast clouds	few clouds	7

Merging temperature and weather description data

In [36]:
```
df_weather_all = pd.merge(df_temp_copy, df_weather_c, suffixes=('_Temp
', '_Des'),\
                          left_index=True, right_index=True, how='oute
r')
```

In [37]:

```
df_weather_all.head()
```

Out[37]:

	datetime_Temp	Portland_Temp	Seattle_Temp	Los Angeles_Temp	Houston_Temp	Chic
1	2012-10-01 13:00:00	48.074000	47.570000	65.696000	59.216000	51.5
2	2012-10-01 14:00:00	48.079854	47.564990	65.692734	59.265636	51.6
3	2012-10-01 15:00:00	48.095360	47.551699	65.683120	59.331818	51.8
4	2012-10-01 16:00:00	48.110866	47.538407	65.673506	59.398000	52.0
5	2012-10-01 17:00:00	48.126372	47.525116	65.663892	59.464182	52.2

5 rows × 30 columns

Bifurcating the weather data city wise

In [38]:

```
Portland_weather=df_weather_all[['datetime_Temp','Portland_Temp','Port
land_bin',\
                        'Portland_Des','Portland_grouping']]
```

In [39]:

```
LosAngeles_weather=df_weather_all[['datetime_Temp','Los Angeles_Temp',
'Los Angeles_bin',\
                        'Los Angeles_Des','LosAngeles_grouping']]
```

In [40]:

```
Houston_weather=df_weather_all[['datetime_Temp','Houston_Temp','Housto
n_bin',\
                        'Houston_Des','Houston_grouping']]
```

In [41]:

```
Chicago_weather=df_weather_all[['datetime_Temp','Chicago_Temp','Chicag
o_bin',\
                        'Chicago_Des','Chicago_grouping']]
```

In [42]:

```
Atlanta_weather=df_weather_all[['datetime_Temp','Atlanta_Temp','Atlant
a_bin',\
                        'Atlanta_Des','Atlanta_grouping']]
```

In [43]:

```
Boston_weather=df_weather_all[['datetime_Temp','Boston_Temp','Boston_b
in',\
                        'Boston_Des','Boston_grouping']]
```

In [44]:

```
Seattle_weather=df_weather_all[['datetime_Temp','Seattle_Temp','Seattl
e_bin',\
                        'Seattle_Des','Seattle_grouping']]
```

In [45]:

```
Portland_weather.head(2)
LosAngeles_weather.head(2)
Houston_weather.head(2)
Chicago_weather.head(2)
Atlanta_weather.head(2)
Boston_weather.head(2)
Seattle_weather.head(2)
```

Out[45]:

	datetime_Temp	Portland_Temp	Portland_bin	Portland_Des	Portland_grouping
1	2012-10-01 13:00:00	48.074000	(40, 50]	scattered clouds	7
2	2012-10-01 14:00:00	48.079854	(40, 50]	scattered clouds	7

Out[45]:

	datetime_Temp	Los Angeles_Temp	Los Angeles_bin	Los Angeles_Des	LosAngeles_grouping
1	2012-10-01 13:00:00	65.696000	(60, 70]	mist	5
2	2012-10-01 14:00:00	65.692734	(60, 70]	sky is clear	6

Out[45]:

	datetime_Temp	Houston_Temp	Houston_bin	Houston_Des	Houston_grouping
1	2012-10-01 13:00:00	59.216000	(50, 60]	sky is clear	6
2	2012-10-01 14:00:00	59.265636	(50, 60]	sky is clear	6

Out[45]:

	datetime_Temp	Chicago_Temp	Chicago_bin	Chicago_Des	Chicago_grouping
1	2012-10-01 13:00:00	51.548000	(50, 60]	overcast clouds	7
2	2012-10-01 14:00:00	51.628444	(50, 60]	overcast clouds	7

Out[45]:

	datetime_Temp	Atlanta_Temp	Atlanta_bin	Atlanta_Des	Atlanta_grouping
1	2012-10-01 13:00:00	69.584000	(60, 70]	light rain	3
2	2012-10-01 14:00:00	69.593615	(60, 70]	overcast clouds	7

Out[45]:

	datetime_Temp	Boston_Temp	Boston_bin	Boston_Des	Boston_grouping
1	2012-10-01 13:00:00	57.236000	(50, 60]	sky is clear	6
2	2012-10-01 14:00:00	57.264966	(50, 60]	few clouds	7

Out[45]:

	datetime_Temp	Seattle_Temp	Seattle_bin	Seattle_Des	Seattle_grouping
1	2012-10-01 13:00:00	47.57000	(40, 50]	sky is clear	6
2	2012-10-01 14:00:00	47.56499	(40, 50]	sky is clear	6

CLEANING OF CRIME DATA

Portland Crime Data

Loading crime data

In [46]:

```
df_crime_2012 = pd.read_csv('P_crime_2012.csv',usecols=['Report Date',
'Report Time', \
                                                         'Major Offens
e Type'])
```

In [47]:

```
df_crime_2013 = pd.read_csv('P_crime_2013.csv',usecols=['Report Date',
'Report Time', \
                                                         'Major Offens
e Type'])
```

In [48]:

```
df_crime_2014 = pd.read_csv('P_crime_2014.csv',usecols=['Report Date',
'Report Time', \
                                                         'Major Offens
e Type'])
```

In [49]:

```
df_crime_2015 = pd.read_csv('P_crime_2015.csv',usecols=['Occur Time',
'Offense Type', \
                                             'Report Date'])
```

```
In [50]: df_crime_2016 = pd.read_csv('P_crime_2016.csv',usecols=['Occur Time',
         'Offense Type', \
                                                            'Report Date'])
```

```
In [51]: df_crime_2017 = pd.read_csv('P_crime_2017.csv',usecols=['Occur Time',
         'Offense Type', \
                                                            'Report Date'])
```

Merging data for 2012-2014 as the column names for(2012-2014) is different from (2015-2017)

```
In [52]: Portland_merge_1 = pd.concat([df_crime_2012, df_crime_2013, df_crime_2
         014], join="outer")
```

```
In [53]: Portland_merge_1.head(2)
```

Out[53]:

	Report Date	Report Time	Major Offense Type
0	03/09/2012	14:08:00	Assault, Simple
1	12/26/2012	21:41:00	Larceny

Checking is it contains any null values

```
In [54]: Portland_merge_1.isna().any()
```

```
Out[54]: Report Date            False
         Report Time            False
         Major Offense Type     False
         dtype: bool
```

Renaming the columns

```
In [55]: Portland_merge_1.columns=['Date','Time','Offense Type']
```

```
In [56]: Portland_merge_1.head(3)
```

Out[56]:

	Date	Time	Offense Type
0	03/09/2012	14:08:00	Assault, Simple
1	12/26/2012	21:41:00	Larceny
2	11/03/2012	21:39:00	Larceny

Merging the crime data for portland from 2015-2017

```
In [57]: Portland_merge_2 = pd.concat([df_crime_2015, df_crime_2016, df_crime_2
         017], join="outer")
```

```
In [58]: Portland_merge_2.head(2)
```

Out[58]:

	Occur Time	Offense Type	Report Date
0	1044	Rape	5/29/15
1	1300	Sodomy	5/15/15

Checking is it contains any null values

```
In [59]: Portland_merge_2.isna().any()
```

```
Out[59]: Occur Time      False
         Offense Type    False
         Report Date     False
         dtype: bool
```

Renaming of columns to make it same as merge_1

```
In [60]: Portland_merge_2.columns=['Time','Offense Type','Date']
```

```
In [61]: Portland_merge_2.head(2)
```

Out[61]:

	Time	Offense Type	Date
0	1044	Rape	5/29/15
1	1300	Sodomy	5/15/15

Time column in Portland_merge_2 is not in proper format, so we are converting it first to datetime and then padding 0's and then rounding it off to the near hour

```
In [62]: Portland_merge_2['Date']=pd.to_datetime(Portland_merge_2['Date'], form
         at='%m/%d/%y')
```

In [63]:

```
merge_10 = Portland_merge_2[Portland_merge_2.Time < 10]
merge_10.head()
```

Out[63]:

	Time	Offense Type	Date
2	0	False Pretenses/Swindle/Confidence Game	2015-05-26
3	0	Identity Theft	2015-05-26
4	1	Identity Theft	2015-05-12
8	0	Identity Theft	2015-05-19
13	0	Hacking/Computer Invasion	2015-05-26

In [64]:

```
merge_10['Dummy'] = '000'
merge_10.Time = merge_10.Dummy + merge_10.Time.map(str)
merge_10.head()
```

Out[64]:

	Time	Offense Type	Date	Dummy
2	0000	False Pretenses/Swindle/Confidence Game	2015-05-26	000
3	0000	Identity Theft	2015-05-26	000
4	0001	Identity Theft	2015-05-12	000
8	0000	Identity Theft	2015-05-19	000
13	0000	Hacking/Computer Invasion	2015-05-26	000

In [65]:

```
merge_10to99 = Portland_merge_2[(Portland_merge_2['Time'] >= 10) & (Po
rtland_merge_2['Time'] <= 99)]
merge_10to99.head()
```

Out[65]:

	Time	Offense Type	Date
433	30	Burglary	2015-05-01
576	30	Theft From Motor Vehicle	2015-05-02
605	35	Vandalism	2015-05-02
659	25	Aggravated Assault	2015-05-03
674	10	Burglary	2015-05-03

In [66]:
```
merge_10to99['Dummy'] = '00'
merge_10to99.Time = merge_10to99.Dummy + merge_10to99.Time.map(str)
merge_10to99.head()
```

Out[66]:

	Time	Offense Type	Date	Dummy
433	0030	Burglary	2015-05-01	00
576	0030	Theft From Motor Vehicle	2015-05-02	00
605	0035	Vandalism	2015-05-02	00
659	0025	Aggravated Assault	2015-05-03	00
674	0010	Burglary	2015-05-03	00

In [67]:
```
merge_99to999 = Portland_merge_2[(Portland_merge_2.Time > 99) & (Portl
and_merge_2.Time <= 999)]
merge_99to999.head()
```

Out[67]:

	Time	Offense Type	Date
5	700	Fondling	2015-05-11
11	825	False Pretenses/Swindle/Confidence Game	2015-05-01
12	825	Identity Theft	2015-05-01
31	100	Identity Theft	2015-05-01
32	100	Theft From Motor Vehicle	2015-05-06

In [68]:
```
merge_99to999['Dummy'] = '0'
merge_99to999.Time = merge_99to999.Dummy + merge_99to999.Time.map(str)
merge_99to999.head(2)
```

Out[68]:

	Time	Offense Type	Date	Dummy
5	0700	Fondling	2015-05-11	0
11	0825	False Pretenses/Swindle/Confidence Game	2015-05-01	0

In [69]:
```
merge_1000 = Portland_merge_2[(Portland_merge_2.Time > 999)]
merge_1000.head(2)
```

Out[69]:

	Time	Offense Type	Date
0	1044	Rape	2015-05-29
1	1300	Sodomy	2015-05-15

```
In [70]: merge_all = pd.concat([merge_10,merge_10to99,merge_99to999,merge_1000
         ], join="outer")
```

```
In [71]: merge_all['Time'] =  pd.to_datetime(merge_all['Time'], format='%H%M')
```

```
In [72]: merge_all.head(2)
```

Out[72]:

	Date	Dummy	Offense Type	Time
2	2015-05-26	000	False Pretenses/Swindle/Confidence Game	1900-01-01
3	2015-05-26	000	Identity Theft	1900-01-01

```
In [73]: merge_all['Time'] = pd.to_datetime(merge_all["Time"]).dt.time
```

```
In [74]: merge_all.head(2)
```

Out[74]:

	Date	Dummy	Offense Type	Time
2	2015-05-26	000	False Pretenses/Swindle/Confidence Game	00:00:00
3	2015-05-26	000	Identity Theft	00:00:00

```
In [75]: merge_all['Date']= merge_all['Date'].astype(str)
```

Dropping column Dummy after formating the time

```
In [76]: merge_all.drop(columns='Dummy',axis=1,inplace=True)
```

```
In [77]: Portland_crime = pd.concat([merge_all, Portland_merge_1], join="outer"
         )
```

```
In [78]: Portland_crime.isna().any()
```

```
Out[78]: Date            False
         Offense Type    False
         Time            False
         dtype: bool
```

Dropping the data with na values

```
In [79]: Portland_crime = Portland_crime.dropna(how='any',axis=0)
In [80]: len(Portland_crime)
```

```
Out[80]: 323488
```

```
In [81]: Portland_crime.head(2)
```

Out[81]:

	Date	Offense Type	Time
2	2015-05-26	False Pretenses/Swindle/Confidence Game	00:00:00
3	2015-05-26	Identity Theft	00:00:00

```
In [82]: Portland_crime['Time'] = Portland_crime['Time'].astype(str)
```

Concatenating Date and time together to create a datetime column

```
In [83]: Portland_crime['datetime'] = Portland_crime['Date'] +' ' + Portland_cr
         ime['Time']
```

```
In [84]: Portland_crime['datetime'] = pd.to_datetime(Portland_crime['datetime']
         )
```

```
In [85]: Portland_crime.head(2)
```

Out[85]:

	Date	Offense Type	Time	datetime
2	2015-05-26	False Pretenses/Swindle/Confidence Game	00:00:00	2015-05-26
3	2015-05-26	Identity Theft	00:00:00	2015-05-26

Rounding up the time values

```
In [86]: Portland_crime['Datetime'] = Portland_crime['datetime'].dt.round('1H')
```

Dropping the columns datetime, Date and Time which are not required after creating'Datetime'

```
In [87]: Portland_crime.drop(columns='datetime',axis=1,inplace=True)
```

```
In [88]: Portland_crime.drop(columns=['Date','Time'],axis=1,inplace=True)
```

In [89]:
```
Portland_crime.head(2)
```

Out[89]:

	Offense Type	Datetime
2	False Pretenses/Swindle/Confidence Game	2015-05-26
3	Identity Theft	2015-05-26

Changing 'Offense Type' to lowercase and cleaning up the field by replacing '/' or '-' with ' '

In [90]:
```
Portland_crime['Offense Type'] = Portland_crime['Offense Type'].str.lower()
Portland_crime['Offense Type'] = Portland_crime['Offense Type'].str.replace('/',' ')
Portland_crime['Offense Type'] = Portland_crime['Offense Type'].str.replace('-',' ')
```

Fetching the year from datetime so as to take data from 2012-2017

In [91]:
```
Portland_crime["Year"] = pd.to_datetime(Portland_crime["Datetime"]).dt.year
```

In [92]:
```
Portland_crime= Portland_crime[Portland_crime.Year.between(2012,2017)]
```

In [93]:
```
Portland_crime.sort_values('Year',ascending=True).head()
```

Out[93]:

	Offense Type	Datetime	Year
24810	larceny	2012-08-21 19:00:00	2012
7436	larceny	2012-02-24 10:00:00	2012
7435	trespass	2012-06-14 16:00:00	2012
7434	burglary	2012-07-08 00:00:00	2012
7433	larceny	2012-09-10 09:00:00	2012

Atlanta Crime data

Loading crime data and using the required columns

```
In [94]: df_AT12 = pd.read_csv('AT-2012.csv',usecols=['occur_date', 'occur_time
         ', \
                                                                'UC2 Literal']
         )
```

```
In [95]: df_AT13 = pd.read_csv('AT-2013.csv',usecols=['occur_date', 'occur_time
         ', \
                                                                 'UC2 Literal'
         ])
```

```
In [96]: df_AT14 = pd.read_csv('AT-2014.csv',usecols=['occur_date', 'occur_time
         ', \
                                                                 'UC2 Literal'
         ])
```

```
In [97]: df_AT15 = pd.read_csv('AT-2015.csv',usecols=['occur_date', 'occur_time
         ', \
                                                                 'UC2 Literal'
         ])
```

```
In [98]: df_AT16 = pd.read_csv('AT-2016.csv',usecols=['occur_date', 'occur_time
         ', \
                                                                 'UC2 Literal'
         ])
```

```
In [99]: df_AT17 = pd.read_csv('AT-2017.csv',usecols=['occur_date', 'occur_time
         ', \
                                                               'UC2 Literal'])
```

```
In [100]: merge_AT = pd.concat([df_AT12, df_AT13, df_AT14,df_AT15,df_AT16,df_AT1
          7], join="outer")
```

```
In [101]: merge_AT.head(2)
```

Out[101]:

	occur_date	occur_time	UC2 Literal
0	01/14/2012	03:00:00	LARCENY-NON VEHICLE
1	08/22/2011	12:00:00	LARCENY-NON VEHICLE

```
In [102]: merge_AT.columns=(['Date','Time','Offense Type'])
```

In [103]:
```
merge_AT.head(2)
```

Out[103]:

	Date	Time	Offense Type
0	01/14/2012	03:00:00	LARCENY-NON VEHICLE
1	08/22/2011	12:00:00	LARCENY-NON VEHICLE

In [104]:
```
merge_AT["Year"] = pd.to_datetime(merge_AT["Date"]).dt.year
```

In [105]:
```
Atlanta_Crime= merge_AT[merge_AT.Year.between(2012,2017)]
```

In [106]:
```
Atlanta_Crime['Date']=pd.to_datetime(Atlanta_Crime['Date'], format='%m
/%d/%Y')
```

In [107]:
```
Atlanta_Crime.head(2)
```

Out[107]:

	Date	Time	Offense Type	Year
0	2012-01-14	03:00:00	LARCENY-NON VEHICLE	2012
5	2012-01-01	00:04:00	AGG ASSAULT	2012

In [108]:
```
Atlanta_Crime['Time'] = Atlanta_Crime['Time'].astype(str)
```

In [109]:
```
Atlanta_Crime['Date'] = Atlanta_Crime['Date'].astype(str)
```

In [110]:
```
Atlanta_Crime['datetime'] = Atlanta_Crime['Date'] +'  ' + Atlanta_Crime
['Time']
```

In [111]:
```
Atlanta_Crime.head(2)
```

Out[111]:

	Date	Time	Offense Type	Year	datetime
0	2012-01-14	03:00:00	LARCENY-NON VEHICLE	2012	2012-01-14 03:00:00
5	2012-01-01	00:04:00	AGG ASSAULT	2012	2012-01-01 00:04:00

In [112]:
```
Atlanta_Crime.isna().any()
```

Out[112]:
```
Date            False
Time            False
Offense Type    False
Year            False
datetime        False
dtype: bool
```

Dropping the data with na values

In [113]:

```
Atlanta_Crime = Atlanta_Crime.dropna(how='any',axis=0)
```

In [114]:

```
Atlanta_Crime.datetime.dtype
```

Out[114]:

```
dtype('O')
```

In [115]:

```
Atlanta_Crime['datetime'] = pd.to_datetime(Atlanta_Crime['datetime'],\
                                           dayfirst=True,errors='coerce')
```

In [116]:

```
Atlanta_Crime['Datetime']=Atlanta_Crime['datetime'].dt.round('1H')
```

In [117]:

```
Atlanta_Crime.head(2)
```

Out[117]:

	Date	Time	Offense Type	Year	datetime	Datetime
0	2012-01-14	03:00:00	LARCENY-NON VEHICLE	2012	2012-01-14 03:00:00	2012-01-14 03:00:00
5	2012-01-01	00:04:00	AGG ASSAULT	2012	2012-01-01 00:04:00	2012-01-01 00:00:00

In [118]:

```
Atlanta_Crime.drop(columns=['datetime','Date','Time'],axis=1,inplace=True)
```

In [119]:

```
Atlanta_Crime.head(2)
```

Out[119]:

	Offense Type	Year	Datetime
0	LARCENY-NON VEHICLE	2012	2012-01-14 03:00:00
5	AGG ASSAULT	2012	2012-01-01 00:00:00

Changing 'Offense Type' to lowercase and cleaning data by replacing '/' or '-' with ' '

In [120]:

```
Atlanta_Crime['Offense Type'] = Atlanta_Crime['Offense Type'].str.lower()
Atlanta_Crime['Offense Type'] = Atlanta_Crime['Offense Type'].str.replace('/',' ')
Atlanta_Crime['Offense Type'] = Atlanta_Crime['Offense Type'].str.replace('-',' ')
```

CHICAGO CRIME

Loading Chicago Data

```
In [121]: Chicago_crime = pd.read_csv('Chicago_Crimes_final.csv', \
                                   usecols=['Date', 'Primary Type', 'Year'])
```

Renaming the columns to generalize it amongst all cities

```
In [122]: Chicago_crime.columns=['Date', 'Offense Type', 'Year']
```

To check the null value existence

```
In [123]: Chicago_crime.isna().any()
```

```
Out[123]: Date            False
          Offense Type    False
          Year            False
          dtype: bool
```

```
In [124]: Chicago_crime['Offense Type'] = Chicago_crime['Offense Type'].str.lowe
          r()
```

Changing the format to datetime

```
In [125]: Chicago_crime['Date']=pd.to_datetime(Chicago_crime['Date'], format='%m
          /%d/%y %H:%M')
```

```
In [126]: Chicago_crime['Datetime']=Chicago_crime['Date'].dt.round('1H')
```

```
In [127]: Chicago_crime.head(2)
```

Out[127]:

	Date	Offense Type	Year	Datetime
0	2016-05-03 23:40:00	battery	2016	2016-05-04 00:00:00
1	2016-05-03 21:40:00	battery	2016	2016-05-03 22:00:00

```
In [128]: Chicago_crime= Chicago_crime[Chicago_crime.Year.between(2012,2017)]
```

```
In [129]: Chicago_crime.drop(columns=['Date'],axis=1,inplace=True)
```

Changing 'Offense Type' to lowercase and cleaning data by replacing '/' or '-' with ' '

```
In [130]: Chicago_crime['Offense Type'] = Chicago_crime['Offense Type'].str.repl
          ace('/',' ')
          Chicago_crime['Offense Type'] = Chicago_crime['Offense Type'].str.repl
          ace('-',' ')
```

BOSTON CRIME

Loading Boston data and using the required columns

```
In [131]: Boston_crime = pd.read_csv('boston_final_v1.csv',encoding = "ISO-8859-
          1", \
                                      usecols=['OCCURRED_ON_DATE','OFFENSE_CODE_GROU
          P'])
```

```
In [132]: Boston_crime[:2]
```

Out[132]:

	OFFENSE_CODE_GROUP	OCCURRED_ON_DATE
0	Confidence Games	2/2/18 5:00
1	Property Found	2/5/18 21:40

Renaming the columns

```
In [133]: Boston_crime.columns=['Offense Type','Date']
```

```
In [134]: Boston_crime[:2]
```

Out[134]:

	Offense Type	Date
0	Confidence Games	2/2/18 5:00
1	Property Found	2/5/18 21:40

Checking null values

```
In [135]: Boston_crime.isna().any()
```

```
Out[135]: Offense Type    False
          Date            False
          dtype: bool
```

```
In [136]: Boston_crime['Date']=pd.to_datetime(Boston_crime['Date'], format='%m/%d/%y %H:%M')
```

Rounding off the datetime to the near hour

```
In [137]: Boston_crime['Datetime']=Boston_crime['Date'].dt.round('1H')
```

Dropping the date column as it is not required now

```
In [138]: Boston_crime.drop(columns=['Date'],axis=1,inplace=True)
```

```
In [139]: Boston_crime.head(2)
```

Out[139]:

	Offense Type	Datetime
0	Confidence Games	2018-02-02 05:00:00
1	Property Found	2018-02-05 22:00:00

```
In [140]: Boston_crime["Year"] = pd.to_datetime(Boston_crime["Datetime"]).dt.year
```

```
In [141]: Boston_crime= Boston_crime[Boston_crime.Year.between(2012,2017)]
```

Changing 'Offense Type' to lowercase and cleaning data by replacing '/' or '-' with ' '

```
In [142]: Boston_crime['Offense Type'] = Boston_crime['Offense Type'].str.lower()
```

```
In [143]: Boston_crime['Offense Type'] = Boston_crime['Offense Type'].str.replace('/',' ')
          Boston_crime['Offense Type'] = Boston_crime['Offense Type'].str.replace('-',' ')
```

SEATTLE CRIME

Loading the seattle data and use the required columns

```
In [144]: Seattle_crime = pd.read_csv('Seattle_final_v1.csv',\
                                     usecols=['Offense Type', 'Report Date'])
```

```
In [145]: Seattle_crime['Report Date']=pd.to_datetime(Seattle_crime['Report Date'], \
                                                          format='%Y-%m-%dT%H:%M:%S'
          )
```

Rounding off the time in Report Date to near hour

```
In [146]: Seattle_crime['new datetime']=Seattle_crime['Report Date'].dt.round('1H')
```

```
In [147]: Seattle_crime.head(2)
```

Out[147]:

	Offense Type	Report Date	new datetime
0	VEH-THEFT-AUTO	2016-08-22 10:20:00	2016-08-22 10:00:00
1	THEFT-LICENSE PLATE	2016-08-27 17:25:00	2016-08-27 17:00:00

Dropping off the report Date column after creating the Datetime

```
In [148]: Seattle_crime.drop(columns='Report Date',axis=1,inplace=True)
```

```
In [149]: Seattle_crime.head(2)
```

Out[149]:

	Offense Type	new datetime
0	VEH-THEFT-AUTO	2016-08-22 10:00:00
1	THEFT-LICENSE PLATE	2016-08-27 17:00:00

Renaming the column

```
In [150]: Seattle_crime.columns=['Offense Type','Datetime']
```

Changing offense type to lower and stripping of '/' and '-'

```
In [151]: Seattle_crime["Year"] = pd.to_datetime(Seattle_crime["Datetime"]).dt.y
          ear
```

```
In [152]: Seattle_crime= Seattle_crime[Seattle_crime.Year.between(2012,2017)]
```

```
In [153]: Seattle_crime['Offense Type'] = Seattle_crime['Offense Type'].str.lowe
          r()
```

```
In [154]: Seattle_crime['Offense Type'] = Seattle_crime['Offense Type'].str.repl
          ace('/',' ')
          Seattle_crime['Offense Type'] = Seattle_crime['Offense Type'].str.repl
          ace('-',' ')
```

Los Angeles Crime

loading crime data

```
In [155]: LA_crime = pd.read_csv('LosAngeles_Crime_10-17.csv',\
                                usecols=['Date Occurred','Time Occurred','Crime Cod
          e Description'])
```

```
In [156]: LA_crime.head(2)
```

```
Out[156]:
```

	Date Occurred	Time Occurred	Crime Code Description
0	03/11/2013	1800	INTIMATE PARTNER - SIMPLE ASSAULT
1	01/22/2010	2300	VEHICLE - STOLEN

Cleaning null values

```
In [157]: LA_crime.isna().any()
```

```
Out[157]: Date Occurred             False
          Time Occurred             False
          Crime Code Description     True
          dtype: bool
```

```
In [158]: LA_crime = LA_crime.dropna(how='any',axis=0)
In [159]: LA_crime.head(2)
```

Out[159]:

	Date Occurred	Time Occurred	Crime Code Description
0	03/11/2013	1800	INTIMATE PARTNER - SIMPLE ASSAULT
1	01/22/2010	2300	VEHICLE - STOLEN

Adjusting the time into datetime format

```
In [160]: LA_merge_10 = LA_crime[LA_crime['Time Occurred'] < 10]
          LA_merge_10.head(2)
```

Out[160]:

	Date Occurred	Time Occurred	Crime Code Description
244	06/30/2013	1	ROBBERY
446	10/20/2013	1	SEXUAL PENTRATION WITH A FOREIGN OBJECT

```
In [161]: LA_merge_10['dummy'] = '000'
          LA_merge_10['Time Occurred']= LA_merge_10.dummy + LA_merge_10['Time Oc
          curred'].map(str)
          LA_merge_10.head(2)
```

Out[161]:

	Date Occurred	Time Occurred	Crime Code Description	dummy
244	06/30/2013	0001	ROBBERY	000
446	10/20/2013	0001	SEXUAL PENTRATION WITH A FOREIGN OBJECT	000

```
In [162]: LA_merge_10to99 = LA_crime[(LA_crime['Time Occurred'] >= 10) & \
                                     (LA_crime['Time Occurred'] <= 99)]
          LA_merge_10to99.head(2)
```

Out[162]:

	Date Occurred	Time Occurred	Crime Code Description
69	02/17/2013	25	INTIMATE PARTNER - SIMPLE ASSAULT
83	03/02/2013	50	OTHER MISCELLANEOUS CRIME

In [163]:

```
LA_merge_10to99['dummy'] = '00'
LA_merge_10to99['Time Occurred'] = LA_merge_10to99.dummy +\
LA_merge_10to99['Time Occurred'].map(str)
LA_merge_10to99.head(2)
```

Out[163]:

	Date Occurred	Time Occurred	Crime Code Description	dummy
69	02/17/2013	0025	INTIMATE PARTNER - SIMPLE ASSAULT	00
83	03/02/2013	0050	OTHER MISCELLANEOUS CRIME	00

In [164]:

```
LA_merge_99to999 = LA_crime[(LA_crime['Time Occurred'] > 99) & \
                            (LA_crime['Time Occurred'] <= 999)]
LA_merge_99to999.head(2)
```

Out[164]:

	Date Occurred	Time Occurred	Crime Code Description
9	01/15/2013	800	EMBEZZLEMENT, GRAND THEFT ($950.01 & OVER)
15	01/01/2013	500	INTIMATE PARTNER - SIMPLE ASSAULT

In [165]:

```
LA_merge_99to999['dummy'] = '0'
LA_merge_99to999['Time Occurred'] = LA_merge_99to999.dummy + \
LA_merge_99to999['Time Occurred'].map(str)
LA_merge_99to999.head(2)
```

Out[165]:

	Date Occurred	Time Occurred	Crime Code Description	dummy
9	01/15/2013	0800	EMBEZZLEMENT, GRAND THEFT ($950.01 & OVER)	0
15	01/01/2013	0500	INTIMATE PARTNER - SIMPLE ASSAULT	0

In [166]:

```
LA_merge_1000 = LA_crime[(LA_crime['Time Occurred'] > 999)]
LA_merge_1000.head(2)
```

Out[166]:

	Date Occurred	Time Occurred	Crime Code Description
0	03/11/2013	1800	INTIMATE PARTNER - SIMPLE ASSAULT
1	01/22/2010	2300	VEHICLE - STOLEN

In [167]:

```
LA_merge_all = pd.concat([LA_merge_10,LA_merge_10to99,LA_merge_99to999
,LA_merge_1000 ],\
                         join="outer")
```

In [168]:

```
LA_crime = LA_merge_all
LA_crime.head(2)
```

Out[168]:

	Crime Code Description	Date Occurred	Time Occurred	dummy
244	ROBBERY	06/30/2013	0001	000
446	SEXUAL PENTRATION WITH A FOREIGN OBJECT	10/20/2013	0001	000

In [169]:

```
LA_crime.drop(columns='dummy',axis=1,inplace=True)
```

In [170]:

```
LA_crime['Time Occurred'] =  pd.to_datetime(LA_crime['Time Occurred'],
format='%H%M')
```

In [171]:

```
LA_crime['Date Occurred'] = LA_crime['Date Occurred'].astype(str)
```

In [172]:

```
LA_crime.head(2)
```

Out[172]:

	Crime Code Description	Date Occurred	Time Occurred
244	ROBBERY	06/30/2013	1900-01-01 00:01:00
446	SEXUAL PENTRATION WITH A FOREIGN OBJECT	10/20/2013	1900-01-01 00:01:00

In [173]:

```
LA_crime['Time Occurred'] = pd.to_datetime(LA_crime["Time Occurred"]).
dt.time
```

In [174]:

```
LA_crime['Time Occurred'] = LA_crime['Time Occurred'].astype(str)
```

In [175]:

```
LA_crime['datetime'] = LA_crime['Date Occurred'] +' ' + LA_crime['Time
Occurred']
```

In [176]:

```
LA_crime['datetime'] = pd.to_datetime(LA_crime['datetime'])
```

In [177]:

```
LA_crime.datetime.dtype
```

Out[177]:

```
dtype('<M8[ns]')
```

In [178]:

```
LA_crime['new datetime']=LA_crime['datetime'].dt.round('1H')
```

```
In [179]: LA_crime.drop(columns=['Date Occurred','Time Occurred','datetime'],axi
          s=1,inplace=True)
```

```
In [180]: LA_crime["Year"] = pd.to_datetime(LA_crime["new datetime"]).dt.year
```

```
In [181]: LA_crime= LA_crime[LA_crime.Year.between(2012,2017)]
```

```
In [182]: LA_crime['Crime Code Description'] = LA_crime['Crime Code Description'
          ].str.lower()
          LA_crime['Crime Code Description'] = LA_crime['Crime Code Description'
          ].str.replace('/',' ')
          LA_crime['Crime Code Description'] = LA_crime['Crime Code Description'
          ].str.replace('-',' ')
```

```
In [183]: LA_crime.columns=['Offense Type','Datetime','Year']
```

```
In [184]: LA_crime.head(2)
```

Out[184]:

	Offense Type	Datetime	Year
244	robbery	2013-06-30	2013
446	sexual pentration with a foreign object	2013-10-20	2013

Houston Crime

In [185]:

```
df_2015_jan= pd.read_csv('2015_January.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_feb= pd.read_csv('2015_February.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_march= pd.read_csv('2015_March.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_apr= pd.read_csv('2015_April.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_may= pd.read_csv('2015_May.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_jun= pd.read_csv('2015_June.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_jul= pd.read_csv('2015_July.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_aug= pd.read_csv('2015_August.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_sep= pd.read_csv('2015_September.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_oct= pd.read_csv('2015_October.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_nov= pd.read_csv('2015_November.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2015_dec= pd.read_csv('2015_December.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_jan= pd.read_csv('2016_January.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_feb= pd.read_csv('2016_February.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_mar= pd.read_csv('2016_March.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_apr= pd.read_csv('2016_April.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_may= pd.read_csv('2016_May.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_jun= pd.read_csv('2016_June.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_jul= pd.read_csv('2016_July.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_aug= pd.read_csv('2016_August.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_nov= pd.read_csv('2016_November.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2016_dec= pd.read_csv('2016_December.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2017_jan= pd.read_csv('2017_January.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2017_feb= pd.read_csv('2017_February.csv',usecols=['Date', 'Hour', 'Offense Type'])
df_2017_mar= pd.read_csv('2017_March.csv',usecols=['Date', 'Hour', 'Offense Type'])
```

Merging the data seperately as each file have different format

In [186]:

```
merge_2015 = pd.concat([df_2015_jan, df_2015_feb, df_2015_march,df_201
5_apr,df_2015_may,df_2015_jun,\
                   df_2015_jul,df_2015_aug,df_2015_sep,df_2015_oct,df_
2015_nov,df_2015_dec], join="outer")
merge_16 = pd.concat([df_2016_jan, df_2016_feb, df_2016_mar,df_2016_ap
r,df_2016_may,df_2016_jun,\
                       df_2016_jul,df_2016_aug], join="outer")
merge_2016=pd.concat([df_2016_nov,df_2016_dec], join="outer")
merge_2017 = pd.concat([df_2017_jan, df_2017_feb, df_2017_mar], join="
outer")
```

In [187]:

```
merge_2017['Date']=pd.to_datetime(merge_2017['Date'],format='%m/%d/%Y'
)

merge_2016['Date']=pd.to_datetime(merge_2016['Date'],format='%m/%d/%Y'
)
```

In [188]:

```
merge_houston_1 = pd.concat([merge_2015, merge_16], join="outer")
```

In [189]:

```
merge_houston_2= pd.concat([merge_2017, merge_2016], join="outer")
```

In [190]:

```
merge_houston_1.head(2)
```

Out[190]:

	Date	Hour	Offense Type
0	1/24/15	8	Auto Theft
1	1/25/15	13	Theft

Adjusting the time into datetime format

In [191]:

```
merge_houston_1_10 = merge_houston_1[merge_houston_1['Hour'] < 10]
merge_houston_1_10['dummy'] = '0'
merge_houston_1_10['Hour']= merge_houston_1_10.dummy + merge_houston_1
_10['Hour'].map(str)
merge_houston_1_10.head(2)
```

Out[191]:

	Date	Hour	Offense Type	dummy
0	1/24/15	08	Auto Theft	0
2	1/26/15	05	Auto Theft	0

In [192]:
```
merge_houston_1_10toall = merge_houston_1[merge_houston_1['Hour'] >= 1
0]
```

In [193]:
```
merge_houston_1_all = pd.concat([merge_houston_1_10,merge_houston_1_10
toall], join="outer")
```

In [194]:
```
merge_houston_1_all.head(2)
```

Out[194]:

	Date	Hour	Offense Type	dummy
0	1/24/15	08	Auto Theft	0
2	1/26/15	05	Auto Theft	0

In [195]:
```
merge_houston_1_all['Hour']=merge_houston_1_all['Hour'].astype(str)
```

In [196]:
```
merge_houston_1_all['datetime'] =  merge_houston_1_all['Date'] + ' ' +
\
merge_houston_1_all['Hour']
```

In [197]:
```
merge_houston_1_all['datetime']=pd.to_datetime(merge_houston_1_all['da
tetime'],\
                                                format='%m/%d/%y %H')
```

In [198]:
```
merge_houston_1_all.head(2)
```

Out[198]:

	Date	Hour	Offense Type	dummy	datetime
0	1/24/15	08	Auto Theft	0	2015-01-24 08:00:00
2	1/26/15	05	Auto Theft	0	2015-01-26 05:00:00

In [199]:
```
merge_houston_2.head(2)
```

Out[199]:

	Date	Hour	Offense Type
0	2017-01-22	12	Theft
1	2017-01-23	9	Theft

```
In [200]: merge_houston_2_10 = merge_houston_2[merge_houston_2['Hour'] < 10]
          merge_houston_2_10['dummy'] = '0'
          merge_houston_2_10['Hour']= merge_houston_2_10.dummy + merge_houston_2
          _10['Hour'].map(str)
          merge_houston_2_10.head(2)
```

Out[200]:

	Date	Hour	Offense Type	dummy
1	2017-01-23	09	Theft	0
6	2017-01-27	08	Burglary	0

```
In [201]: merge_houston_2_10toall = merge_houston_2[merge_houston_2['Hour'] >= 1
          0]
```

```
In [202]: merge_houston_2_all = pd.concat([merge_houston_2_10,merge_houston_2_10
          toall], join="outer")
```

```
In [203]: merge_houston_2_all['Hour']=merge_houston_2_all['Hour'].astype(str)
          merge_houston_2_all['Date']=merge_houston_2_all['Date'].astype(str)
```

```
In [204]: merge_houston_2_all['Hour']= merge_houston_2_all['Hour'].replace('24',
          '00')
```

```
In [205]: merge_houston_2_all['datetime'] = merge_houston_2_all['Date'] +' ' +\
          merge_houston_2_all['Hour']
```

```
In [206]: merge_houston_2_all['datetime']=pd.to_datetime(merge_houston_2_all['da
          tetime'],\
                                                        format='%Y-%m-%d %H')
```

```
In [207]: merge_houston_2_all.head(2)
```

Out[207]:

	Date	Hour	Offense Type	dummy	datetime
1	2017-01-23	09	Theft	0	2017-01-23 09:00:00
6	2017-01-27	08	Burglary	0	2017-01-27 08:00:00

Concatinating all the files after changing them into appropriate formats

```
In [208]: Houston_crime= pd.concat([merge_houston_1_all, merge_houston_2_all], j
          oin="outer")
```

```
In [209]: Houston_crime['Offense Type'] = Houston_crime['Offense Type'].str.lower()
          Houston_crime['Offense Type']  = Houston_crime['Offense Type'] .str.replace('/',' ')
          Houston_crime['Offense Type']  = Houston_crime['Offense Type'] .str.replace('-',' ')
```

Removing inappropriate Offense type descriptions

```
In [210]: Houston_crime=Houston_crime[Houston_crime['Offense Type'] != '1']
```

```
In [211]: Houston_crime.head(2)
```

Out[211]:

	Date	Hour	Offense Type	dummy	datetime
0	1/24/15	08	auto theft	0	2015-01-24 08:00:00
2	1/26/15	05	auto theft	0	2015-01-26 05:00:00

```
In [212]: Houston_crime['new datetime']=Houston_crime['datetime'].dt.round('1H')
```

```
In [213]: Houston_crime.drop(columns=['datetime','Hour','Date','dummy'],axis=1,inplace=True)
```

```
In [214]: Houston_crime.head(2)
```

Out[214]:

	Offense Type	new datetime
0	auto theft	2015-01-24 08:00:00
2	auto theft	2015-01-26 05:00:00

```
In [215]: Houston_crime.columns=['Offense Type','Datetime']
```

```
In [216]: Houston_crime["Year"] = pd.to_datetime(Houston_crime["Datetime"]).dt.year
```

COnsidering data from 2012 to 2017

```
In [217]: Houston_crime= Houston_crime[Houston_crime.Year.between(2012,2017)]
In [218]: Houston_crime.head(2)
```

Out[218]:

	Offense Type	Datetime	Year
0	auto theft	2015-01-24 08:00:00	2015
2	auto theft	2015-01-26 05:00:00	2015

Classifying Offenses into categories and numering them

```
In [219]: string_theft = ["theft",'larceny','shoplifting','stolen','picking','ro
bbery','purse','stolen','pickpocket']
string_burglary = ['burglary','home invasion']
string_assault = ['asslt','simple assault','aggravated assault','assau
lt','agg','battery']
string_sex_crimes = ['rape','sodomy','fondling','sexual','prostitution
','pornography','incest','sex','obscenity',\
                     'public indecency','indecent','incest','pandering
','lewd','pimping']
string_forgery = ['forgery']
string_weapons = ['weapons','concealed carry license violation','firea
rm','ballistics','explosives','shots','bomb']
string_drug = ['drug','narcotic','narcotics','narc']
string_fraud = ['false pretenses','swindle','confidence game', 'confid
ence games',\
                'embezzlement','embezzle','fraud','extortion','imperso
nation','deceptive','counterfeiting', \
                'extortion', 'forgery','bribery','counterfeit']
string_arson = ['arson']
string_kidnapping=['kidnapping','abduction','kidnap','trafficking']
string_homicide=['murder','manslaughter','homicide','lynching']
string_harass = ['harassment', 'threats','threatening','stalking']
string_driving = ['operating under the influence', 'duii', 'dui', 'lic
ense violation','reckless driving']
```

In [220]:

```
def crime_type(crime_col):
    for item in string_theft:
        if item in crime_col:
            return 1
    for item in string_burglary:
        if item in crime_col:
            return 2
    for item in string_assault:
        if item in crime_col:
            return 3
    for item in string_sex_crimes:
        if item in crime_col:
            return 4
    for item in string_forgery:
        if item in crime_col:
            return 5
    for item in string_weapons:
        if item in crime_col:
            return 6
    for item in string_drug:
        if item in crime_col:
            return 7
    for item in string_fraud:
        if item in crime_col:
            return 8
    for item in string_arson:
        if item in crime_col:
            return 9
    for item in string_kidnapping:
        if item in crime_col:
            return 10
    for item in string_homicide:
        if item in crime_col:
            return 11
    for item in string_harass:
        if item in crime_col:
            return 12
    for item in string_driving:
        if item in crime_col:
            return 13
        else:
            return 14
```

In [221]:

```
Portland_crime['Offense_Num'] = Portland_crime['Offense Type'].apply(c
rime_type)
```

In [222]:

```
Atlanta_Crime['Offense_Num'] = Atlanta_Crime['Offense Type'].apply(cri
me_type)
```

In [223]:

```
Chicago_crime['Offense_Num'] = Chicago_crime['Offense Type'].apply(cri
me_type)
```

In [224]:

```
Boston_crime['Offense_Num'] = Boston_crime['Offense Type'].apply(crime
_type)
```

In [225]:

```
Seattle_crime['Offense_Num'] = Seattle_crime['Offense Type'].apply(cri
me_type)
```

In [226]:

```
Houston_crime['Offense_Num'] = Houston_crime['Offense Type'].apply(cri
me_type)
```

In [227]:

```
LA_crime['Offense Type']=LA_crime['Offense Type'].astype(str)
```

In [228]:

```
LA_crime['Offense_Num'] = (LA_crime['Offense Type']).apply(crime_type)
```

In [229]:

```
Portland_crime.head(2)
```

Out[229]:

	Offense Type	Datetime	Year	Offense_Num
2	false pretenses swindle confidence game	2015-05-26	2015	8
3	identity theft	2015-05-26	2015	1

In [230]:

```
Atlanta_Crime.head(2)
```

Out[230]:

	Offense Type	Year	Datetime	Offense_Num
0	larceny non vehicle	2012	2012-01-14 03:00:00	1
5	agg assault	2012	2012-01-01 00:00:00	3

In [231]:

```
Chicago_crime.head(2)
```

Out[231]:

	Offense Type	Year	Datetime	Offense_Num
0	battery	2016	2016-05-04 00:00:00	3
1	battery	2016	2016-05-03 22:00:00	3

In [232]:

```
Boston_crime.head(2)
```

Out[232]:

	Offense Type	Datetime	Year	Offense_Num
60	fraud	2017-12-01 09:00:00	2017	8
98	property lost	2016-08-01 16:00:00	2016	14

```
In [233]: Seattle_crime.head(2)
```

Out[233]:

	Offense Type	Datetime	Year	Offense_Num
0	veh theft auto	2016-08-22 10:00:00	2016.0	1
1	theft license plate	2016-08-27 17:00:00	2016.0	1

```
In [234]: Houston_crime.head(2)
```

Out[234]:

	Offense Type	Datetime	Year	Offense_Num
0	auto theft	2015-01-24 08:00:00	2015	1
2	auto theft	2015-01-26 05:00:00	2015	1

```
In [235]: LA_crime.head(2)
```

Out[235]:

	Offense Type	Datetime	Year	Offense_Num
244	robbery	2013-06-30	2013	1
446	sexual pentration with a foreign object	2013-10-20	2013	4

Merging Crime and Weather Data

Boston

```
In [236]: Boston_weather.datetime_Temp.dtype

Out[236]: dtype('O')
```

```
In [237]: Boston_weather['datetime_Temp']=pd.to_datetime(Boston_weather['datetim
          e_Temp'],\
                                                                format='%Y/%m/%d
          %H:%M:%S')
```

```
In [238]: Boston_weather.datetime_Temp.dtype

Out[238]: dtype('<M8[ns]')
```

```
In [239]: Boston_crime.Datetime.dtype

Out[239]: dtype('<M8[ns]')
```

In [240]:
```
Boston_merged = pd.merge(Boston_weather, Boston_crime, left_on = 'date
time_Temp',\
                                  right_on = 'Datetime', how='right')
```

In [241]:
```
Boston_merged.head(2)
```

Out[241]:

	datetime_Temp	Boston_Temp	Boston_bin	Boston_Des	Boston_grouping	Offen Ty
0	2015-06-15	60.952634	(60, 70]	scattered clouds	7.0	harassm
1	2015-06-15	60.952634	(60, 70]	scattered clouds	7.0	property lost

In [242]:
```
Boston_merged.drop('datetime_Temp',axis=1,inplace=True)
```

In [243]:
```
Boston_merged.columns= ['Temp','Bins','Temp_Des','TD_num','Offense_Typ
e','Datetime',\
                              'Year','Offense_Num']
```

In [244]:
```
Boston_merged['city'] = 'Boston'
```

In [245]:
```
Boston_merged.head(2)
```

Out[245]:

	Temp	Bins	Temp_Des	TD_num	Offense_Type	Datetime	Year	Offense_Num
0	60.952634	(60, 70]	scattered clouds	7.0	harassment	2015-06-15	2015	12
1	60.952634	(60, 70]	scattered clouds	7.0	property lost	2015-06-15	2015	14

Seattle

In [246]:
```
Seattle_weather.datetime_Temp.dtype
```

Out[246]:
```
dtype('O')
```

In [247]:
```
Seattle_weather['datetime_Temp']=pd.to_datetime(Seattle_weather['datet
ime_Temp'],\
                                                               format='%Y/%m/%d
%H:%M:%S')
```

```
In [248]: Seattle_weather.datetime_Temp.dtype

Out[248]: dtype('<M8[ns]')
```

```
In [249]: Seattle_crime.Datetime.dtype

Out[249]: dtype('<M8[ns]')
```

```
In [250]: Seattle_merged = pd.merge(Seattle_weather, Seattle_crime, left_on = 'd
          atetime_Temp',\
                                    right_on = 'Datetime', how='right')
```

```
In [251]: Seattle_merged.head(2)
```

Out[251]:

	datetime_Temp	Seattle_Temp	Seattle_bin	Seattle_Des	Seattle_grouping	Offense Type
0	2012-10-01 13:00:00	47.57	(40, 50]	sky is clear	6.0	theft carprowl
1	2012-10-01 13:00:00	47.57	(40, 50]	sky is clear	6.0	theft carprowl

```
In [252]: Seattle_merged.drop('datetime_Temp',axis=1,inplace=True)
```

```
In [253]: Seattle_merged.columns= ['Temp','Bins','Temp_Des','TD_num','Offense_Ty
          pe',\
                                  'Datetime','Year','Offense_Num']
```

```
In [254]: Seattle_merged['city'] = 'Seattle'
```

```
In [255]: Seattle_merged.head(2)
```

Out[255]:

	Temp	Bins	Temp_Des	TD_num	Offense_Type	Datetime	Year	Offense_Num	
0	47.57	(40, 50]	sky is clear	6.0	theft carprowl	2012-10-01 13:00:00	2012.0	1	Sea
1	47.57	(40, 50]	sky is clear	6.0	theft carprowl	2012-10-01 13:00:00	2012.0	1	Sea

Atlanta

```
In [256]: Atlanta_weather.datetime_Temp.dtype
```

```
Out[256]: dtype('O')
```

```
In [257]: Atlanta_weather['datetime_Temp']=pd.to_datetime(Atlanta_weather['datet
          ime_Temp'],\
                                                                format='%Y/%m/%d
          %H:%M:%S')
```

```
In [258]: Atlanta_weather.datetime_Temp.dtype
```

```
Out[258]: dtype('<M8[ns]')
```

```
In [259]:  Atlanta_Crime.Datetime.dtype
```

```
Out[259]: dtype('<M8[ns]')
```

```
In [260]: Atlanta_merged = pd.merge(Atlanta_weather, Atlanta_Crime, left_on = 'd
          atetime_Temp',\
                                    right_on = 'Datetime', how='right')
```

```
In [261]: Atlanta_merged.head(2)
```

Out[261]:

	datetime_Temp	Atlanta_Temp	Atlanta_bin	Atlanta_Des	Atlanta_grouping	Offense Type
0	2012-10-01 13:00:00	69.584	(60, 70]	light rain	3.0	larceny from vehicle
1	2012-10-01 13:00:00	69.584	(60, 70]	light rain	3.0	larceny non vehicle

```
In [262]: Atlanta_merged.drop('datetime_Temp',axis=1,inplace=True)
```

```
In [263]: Atlanta_merged.columns= ['Temp','Bins','Temp_Des','TD_num','Offense_Ty
          pe',\
                                  'Year','Datetime','Offense_Num']
```

```
In [264]: Atlanta_merged['city'] = 'Atlanta'
In [265]: Atlanta_merged.head(2)
```

Out[265]:

	Temp	Bins	Temp_Des	TD_num	Offense_Type	Year	Datetime	Offense_Num	c
0	69.584	(60, 70]	light rain	3.0	larceny from vehicle	2012	2012-10-01 13:00:00	1	Atla
1	69.584	(60, 70]	light rain	3.0	larceny non vehicle	2012	2012-10-01 13:00:00	1	Atla

Portland

```
In [266]: Portland_weather.datetime_Temp.dtype

Out[266]: dtype('O')
```

```
In [267]: Portland_weather['datetime_Temp']=pd.to_datetime(Portland_weather['dat
          etime_Temp'],\
                                                            format='%Y/%m/%d
          %H:%M:%S')
```

```
In [268]: Portland_weather.datetime_Temp.dtype

Out[268]: dtype('<M8[ns]')
```

```
In [269]: Portland_crime.Datetime.dtype

Out[269]: dtype('<M8[ns]')
```

```
In [270]: Portland_merged = pd.merge(Portland_weather, Portland_crime, left_on =
          'datetime_Temp',\
                                     right_on = 'Datetime', how='right')
```

```
In [271]: Portland_merged.head(2)
```

Out[271]:

	datetime_Temp	Portland_Temp	Portland_bin	Portland_Des	Portland_grouping	
0	2012-10-01 13:00:00	48.074	(40, 50]	scattered clouds	7.0	larc
1	2012-10-01 13:00:00	48.074	(40, 50]	scattered clouds	7.0	agg ass

```
In [272]: Portland_merged.drop('datetime_Temp',axis=1,inplace=True)
```

```
In [273]: Portland_merged.columns= ['Temp','Bins','Temp_Des','TD_num','Offense_T
          ype',\
                                    'Datetime','Year','Offense_Num']
```

```
In [274]: Portland_merged['city'] = 'Portland'
```

```
In [275]: Portland_merged.head(2)
```

Out[275]:

	Temp	Bins	Temp_Des	TD_num	Offense_Type	Datetime	Year	Offense_Num	
0	48.074	(40, 50]	scattered clouds	7.0	larceny	2012-10-01 13:00:00	2012	1	Port
1	48.074	(40, 50]	scattered clouds	7.0	aggravated assault	2012-10-01 13:00:00	2012	3	Port

Los Angeles

```
In [276]: LosAngeles_weather.datetime_Temp.dtype

Out[276]: dtype('O')
```

```
In [277]: LosAngeles_weather['datetime_Temp']=pd.to_datetime(LosAngeles_weather[
          'datetime_Temp'],\
                                                                  format='%Y/%m/%d
          %H:%M:%S')
```

```
In [278]: LosAngeles_weather.datetime_Temp.dtype
```

```
Out[278]: dtype('<M8[ns]')
```

```
In [279]: LA_crime.Datetime.dtype
```

```
Out[279]: dtype('<M8[ns]')
```

```
In [280]: LosAngeles_merged = pd.merge(LosAngeles_weather, LA_crime, left_on = '
          datetime_Temp',\
                                        right_on = 'Datetime', how='right')
```

```
In [281]: LosAngeles_merged.head(2)
```

Out[281]:

	datetime_Temp	Los Angeles_Temp	Los Angeles_bin	Los Angeles_Des	LosAngeles_grouping	C
0	2012-10-01 13:00:00	65.696	(60, 70]	mist	5.0	v o o
1	2012-10-01 13:00:00	65.696	(60, 70]	mist	5.0	tl ic

```
In [282]: LosAngeles_merged.drop('datetime_Temp',axis=1,inplace=True)
```

```
In [283]: LosAngeles_merged.columns= ['Temp','Bins','Temp_Des','TD_num','Offense
          _Type',\
                                        'Datetime','Year','Offense_Num']
```

```
In [284]: LosAngeles_merged['city'] = 'Los Angeles'
```

```
In [285]: LosAngeles_merged.head(2)
```

Out[285]:

	Temp	Bins	Temp_Des	TD_num	Offense_Type	Datetime	Year	Offense_Num	
0	65.696	(60, 70]	mist	5.0	violation of court order	2012-10-01 13:00:00	2012	14	Los Ang
1	65.696	(60, 70]	mist	5.0	theft of identity	2012-10-01 13:00:00	2012	1	Los Ang

Houston

```
In [286]: Houston_weather.datetime_Temp.dtype

Out[286]: dtype('O')
```

```
In [287]: Houston_weather['datetime_Temp']=pd.to_datetime(Houston_weather['datet
          ime_Temp'],\
                                                                format='%Y/%m/%d
          %H:%M:%S')
```

```
In [288]: Houston_crime.Datetime.dtype

Out[288]: dtype('<M8[ns]')
```

```
In [289]: Houston_merged = pd.merge(Houston_weather, Houston_crime, left_on = 'd
          atetime_Temp',\
                                    right_on = 'Datetime', how='right')
```

```
In [290]: Houston_merged.head(2)
```

Out[290]:

	datetime_Temp	Houston_Temp	Houston_bin	Houston_Des	Houston_grouping	
0	2012-10-01 22:00:00	59.795091	(50, 60]	scattered clouds	7.0	agg ass
1	2012-10-15 14:00:00	76.118000	(70, 80]	broken clouds	7.0	rap

```
In [291]: Houston_merged['city'] = 'Houston'
```

```
In [292]: Houston_merged.drop('datetime_Temp',axis=1,inplace=True)
```

```
In [293]: Houston_merged.columns= ['Temp','Bins','Temp_Des','TD_num','Offense_Ty
          pe',\
                                  'Datetime','Year','Offense_Num','city']
```

In [294]:
```
Houston_merged.head(2)
```

Out[294]:

	Temp	Bins	Temp_Des	TD_num	Offense_Type	Datetime	Year	Offense_Num
0	59.795091	(50, 60]	scattered clouds	7.0	aggravated assault	2012-10-01 22:00:00	2012	3
1	76.118000	(70, 80]	broken clouds	7.0	rape	2012-10-15 14:00:00	2012	4

Chicago

In [295]:
```
Chicago_weather.datetime_Temp.dtype
```

Out[295]:
```
dtype('O')
```

In [296]:
```
Chicago_weather['datetime_Temp']=pd.to_datetime(Chicago_weather['datet
ime_Temp'],\
                                                        format='%Y/%m/%d
%H:%M:%S')
```

In [297]:
```
Chicago_weather.datetime_Temp.dtype
```

Out[297]:
```
dtype('<M8[ns]')
```

In [298]:
```
Chicago_crime.Datetime.dtype
```

Out[298]:
```
dtype('<M8[ns]')
```

In [299]:
```
Chicago_merged = pd.merge(Chicago_weather,Chicago_crime, left_on = 'da
tetime_Temp',\
                                 right_on = 'Datetime', how='right')
```

In [300]:

```
Chicago_merged.head(2)
```

Out[300]:

	datetime_Temp	Chicago_Temp	Chicago_bin	Chicago_Des	Chicago_grouping	Off
0	2012-10-01 13:00:00	51.548	(50, 60]	overcast clouds	7.0	narc
1	2012-10-01 13:00:00	51.548	(50, 60]	overcast clouds	7.0	narc

In [301]:

```
Chicago_merged['city'] = 'Chicago'
```

In [302]:

```
Chicago_merged.drop('datetime_Temp',axis=1,inplace=True)
```

In [303]:

```
Chicago_merged.columns= ['Temp','Bins','Temp_Des','TD_num','Offense_Ty
pe',\
                                'Year','Datetime','Offense_Num','city']
```

In [304]:

```
Chicago_merged.head(2)
```

Out[304]:

	Temp	Bins	Temp_Des	TD_num	Offense_Type	Year	Datetime	Offense_Num	
0	51.548	(50, 60]	overcast clouds	7.0	narcotics	2012	2012-10-01 13:00:00	7	Chic
1	51.548	(50, 60]	overcast clouds	7.0	narcotics	2012	2012-10-01 13:00:00	7	Chic

Merge all cities

In [305]:

```
Merge_City= pd.concat([Portland_merged,Seattle_merged,Atlanta_merged,H
ouston_merged,\
                           LosAngeles_merged,Boston_merged,Chicago_merged],
join="outer")
```

In [306]:

```
Merge_City.head()
```

Out[306]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
1	(40, 50]	2012-10-01 13:00:00	3	aggravated assault	7.0	48.074	scattered clouds	2012.0	Po
2	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
3	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
4	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po

First Insight : Temperature Vs Crime

Rising temperatures in Boston seem to cause a very negative reaction in its population as there is a significant rise in average number of crimes per day; however, in Seattle, people seem to have the opposite response to rising temperatures and crime decreases.

Counting the number of days in each temperature bin for each city

In [307]:

```
Portland_weather_cnt = Portland_weather.groupby('Portland_bin').agg({'
datetime_Temp':'nunique'}).\
rename(columns = {'datetime_Temp':'count'})
```

In [308]:

```
Portland_weather_cnt['city'] = 'Portland'
```

In [309]:
```
Portland_weather_cnt.head()
```

Out[309]:

	count	city
Portland_bin		
(-20, -10]	0	Portland
(-10, 0]	0	Portland
(0, 10]	0	Portland
(10, 20]	115	Portland
(20, 30]	1098	Portland

In [310]:
```
LA_weather_cnt = LosAngeles_weather.groupby('Los Angeles_bin').agg({'datetime_Temp':'nunique'}).\
rename(columns = {'datetime_Temp':'count'})
```

In [311]:
```
LA_weather_cnt['city'] = 'Los Angeles'
```

In [312]:
```
Houston_weather_cnt = Houston_weather.groupby('Houston_bin').agg({'datetime_Temp':'nunique'}).\
rename(columns = {'datetime_Temp':'count'})
```

In [313]:
```
Houston_weather_cnt['city'] = 'Houston'
```

In [314]:
```
Chicago_weather_cnt = Chicago_weather.groupby('Chicago_bin').agg({'datetime_Temp':'nunique'}).\
rename(columns = {'datetime_Temp':'count'})
```

In [315]:
```
Chicago_weather_cnt['city'] = 'Chicago'
```

In [316]:
```
Atlanta_weather_cnt = Atlanta_weather.groupby('Atlanta_bin').agg({'datetime_Temp':'nunique'}).\
rename(columns = {'datetime_Temp':'count'})
```

In [317]:
```
Atlanta_weather_cnt['city'] = 'Atlanta'
```

In [318]:
```
Boston_weather_cnt = Boston_weather.groupby('Boston_bin').agg({'datetime_Temp':'nunique'}).\
rename(columns = {'datetime_Temp':'count'})
```

In [319]:
```
Boston_weather_cnt['city'] = 'Boston'
```

```
In [320]: Seattle_weather_cnt = Seattle_weather.groupby('Seattle_bin').agg({'dat
          etime_Temp':'nunique'}).\
          rename(columns = {'datetime_Temp':'count'})
```

```
In [321]: Seattle_weather_cnt['city'] = 'Seattle'
```

Merging all the cities count files

```
In [322]: temp_cnts = pd.concat([Portland_weather_cnt, LA_weather_cnt, Houston_w
          eather_cnt, Chicago_weather_cnt,\
                                  Atlanta_weather_cnt, Boston_weather_cnt, Seatt
          le_weather_cnt], join="outer", axis=0)
```

```
In [323]: temp_cnts = temp_cnts.reset_index().rename(columns = {'index':'Bins'})
```

```
In [324]: temp_cnts.head()
```

Out[324]:

	Bins	count	city
0	(-20, -10]	0	Portland
1	(-10, 0]	0	Portland
2	(0, 10]	0	Portland
3	(10, 20]	115	Portland
4	(20, 30]	1098	Portland

Loading Population data to calculate crime rate for each city

```
In [325]: Population = pd.read_csv('Population.csv',usecols=['CITY','population'
          ])
```

```
In [326]: Population
```

Out[326]:

	CITY	population
0	Los Angeles	3792621
1	Chicago	2695598
2	Boston	617594
3	Portland	583776
4	Atlanta	420003
5	Houston	2099451
6	Seattle	608660

```
In [327]: LA= 3792621
          Chi = 2695598
          Bos = 617594
          Hous = 2099451
          Port = 583776
          Sea =608660
          AT = 420003
```

```
In [328]: def pops(row):
              if row == 'Chicago':
                  return Chi
              elif row == 'Los Angeles':
                  return LA
              elif row =='Houston':
                  return Hous
              elif row == 'Boston':
                  return Bos
              elif row == 'Portland':
                  return Port
              elif row == 'Seattle':
                  return Sea
              else:
                  return AT
```

Applying groupby for city and temperature bins inorder to find Offense count

```
In [329]: City_comparison_temp = Merge_City.groupby(['city','Bins']).agg({'Offen
          se_Num':'count'}).\
          rename(columns={'Offense_Num':'crime_count'}).reset_index()
```

```
In [330]: City_comparison_temp['total_pop'] = City_comparison_temp['city'].apply(pops)
```

Merging temperature daily count table and city offense count tables

```
In [331]: City_comparison_temp = pd.merge(temp_cnts, City_comparison_temp, left_on = ['Bins','city'], right_on=['Bins', 'city'])
```

```
In [332]: City_comparison_temp.head()
```

Out[332]:

	Bins	count	city	crime_count	total_pop
0	(10, 20]	115	Portland	567	583776
1	(20, 30]	1098	Portland	5968	583776
2	(30, 40]	5676	Portland	33328	583776
3	(40, 50]	12798	Portland	72684	583776
4	(50, 60]	12720	Portland	80748	583776

Calculating crime rate

```
In [333]: City_comparison_temp['crime_rate'] = (City_comparison_temp['crime_count'] / City_comparison_temp['total_pop']) * 100000
```

Calculating average crime rate per day

```
In [334]: City_comparison_temp['crime_rate_avg'] = (City_comparison_temp['crime_rate'] / City_comparison_temp['count']) * 24
```

```
In [335]: City_comparison_temp.head()
```

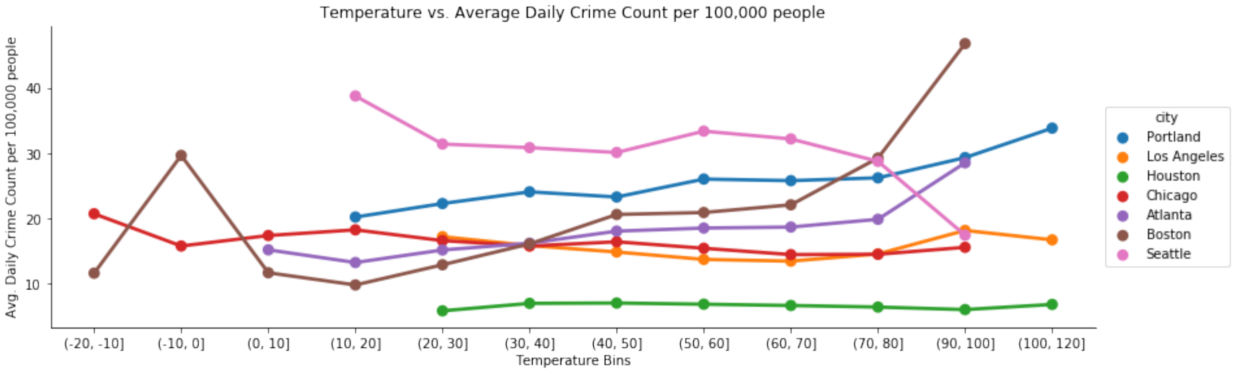
Out[335]:

	Bins	count	city	crime_count	total_pop	crime_rate	crime_rate_avg
0	(10, 20]	115	Portland	567	583776	97.126295	20.269835
1	(20, 30]	1098	Portland	5968	583776	1022.309927	22.345572
2	(30, 40]	5676	Portland	33328	583776	5709.039083	24.139700
3	(40, 50]	12798	Portland	72684	583776	12450.666009	23.348647
4	(50, 60]	12720	Portland	80748	583776	13832.017760	26.098147

Graph: Temperature vs. Average Daily Crime Count per 100,000 people

```
In [336]: sns.factorplot(y = 'crime_rate_avg',data = City_comparison_temp , hue
          = 'city',x='Bins',\
                         aspect=3).set(xlabel = 'Temperature Bins',ylabel ='Avg.
          Daily Crime Count per 100,000 people',\
                                      title='Temperature vs. Average Daily Crime
          Count per 100,000 people')
```

```
Out[336]: <seaborn.axisgrid.FacetGrid at 0x1a1ecfc518>
```



Managerial Insight:

Since weather is usually forecast over a 10-day period with high accuracy, police departments in cities like Boston and Seattle can more efficiently allocate police forces depending on predicted temperatures. This prevents over and under-staffing and can lead to decreased costs for the departments over time.

Machine Learning:

Decision Tree Regressor

```
In [337]: Merge_City_I1 = Merge_City.copy()
```

```
In [338]: Merge_City_I1= Merge_City_I1.dropna(how='any',axis=0)
```

Getting date from datetime

```
In [339]: Merge_City_I1 ['Date'] = pd.to_datetime(Merge_City_I1 ['Datetime']).dt
          .date
```

Getting time from datetime

```
In [340]: Merge_City_I1 ['Time'] = pd.to_datetime(Merge_City_I1 ['Datetime']).dt
          .time
```

```
In [341]: Merge_City_I1.head()
```

Out[341]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
1	(40, 50]	2012-10-01 13:00:00	3	aggravated assault	7.0	48.074	scattered clouds	2012.0	Po
2	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
3	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
4	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po

Assigning number to temperatures for every 20 degrees

In [342]:

```python
def tempseries(row):
    if (row >= -20) & (row < 0):
        return 1
    elif (row >= 0) & (row < 21):
        return 2
    elif (row >= 21) & (row < 40):
        return 3
    elif (row >= 40) & (row < 60):
        return 4
    elif (row >= 60) & (row < 80):
        return 5
    elif (row >= 80) & (row < 100):
        return 6
    elif (row >= 100) & (row < 120):
        return 7
```

In [343]:

```python
Merge_City_I1['Temp']= Merge_City_I1.Temp.apply(tempseries)
```

In [344]:

```python
Merge_City_I1.drop(['Offense_Type','Temp_Des','Datetime','Year','Bins'
],axis=1,inplace=True)
```

In [345]:

```python
Merge_City_I1.head()
```

Out[345]:

	Offense_Num	TD_num	Temp	city	Date	Time
0	1	7.0	4	Portland	2012-10-01	13:00:00
1	3	7.0	4	Portland	2012-10-01	13:00:00
2	1	7.0	4	Portland	2012-10-01	13:00:00
3	1	7.0	4	Portland	2012-10-01	13:00:00
4	1	7.0	4	Portland	2012-10-01	13:00:00

Changing the format of date and time

In [346]:

```python
Merge_City_I1['Time']=pd.to_datetime(Merge_City_I1['Time'],format = '%
H:%M:%S' )
```

In [347]:

```python
Merge_City_I1['Date']=pd.to_datetime(Merge_City_I1['Date'],format = '%
Y-%m-%d' )
```

Assigning numbers for time with an interval of 6 hours

In [348]:

```
def timeseries(row):
    if (row >= '01') & (row < '07'):
        return 1
    elif (row >= '07') & (row < '13'):
        return 2
    elif (row >= '13') & (row < '19'):
        return 3
    elif (row >= '19') & (row < '25'):
        return 4
```

In [349]:

```
Merge_City_I1['Time']=pd.to_datetime(Merge_City_I1['Time']).dt.strftime('%H')
```

In [350]:

```
Merge_City_I1['Month'] = pd.to_datetime(Merge_City_I1['Date']).dt.month
```

In [351]:

```
Merge_City_I1['Day'] = pd.to_datetime(Merge_City_I1['Date']).dt.day
```

In [352]:

```
Merge_City_I1['Time'] = Merge_City_I1['Time'].replace('00','24')
```

In [353]:

```
Merge_City_I1['Time']= Merge_City_I1.Time.apply(timeseries)
```

In [354]:

```
Merge_City_I1.drop(['Date'],axis=1,inplace=True)
```

In [355]:

```
Merge_City_I1.head()
```

Out[355]:

	Offense_Num	TD_num	Temp	city	Time	Month	Day
0	1	7.0	4	Portland	3	10	1
1	3	7.0	4	Portland	3	10	1
2	1	7.0	4	Portland	3	10	1
3	1	7.0	4	Portland	3	10	1
4	1	7.0	4	Portland	3	10	1

In [356]:

```
Merge_City_I1 =  pd.get_dummies(Merge_City_I1, columns = ['city'])
```

In [357]:
```
Merge_City_I1.head()
```

Out[357]:

	Offense_Num	TD_num	Temp	Time	Month	Day	city_Atlanta	city_Boston	city_Chi
0	1	7.0	4	3	10	1	0	0	0
1	3	7.0	4	3	10	1	0	0	0
2	1	7.0	4	3	10	1	0	0	0
3	1	7.0	4	3	10	1	0	0	0
4	1	7.0	4	3	10	1	0	0	0

In [358]:
```
Merge_City_I1 = Merge_City_I1.dropna(how='any',axis=0)
```

In [359]:
```
X=Merge_City_I1.drop(['Offense_Num'], axis=1)
```

In [360]:
```
Y=Merge_City_I1.Offense_Num
```

In [361]:
```
import sklearn as sk
import sklearn.tree as tree
from IPython.display import Image
import pydotplus
```

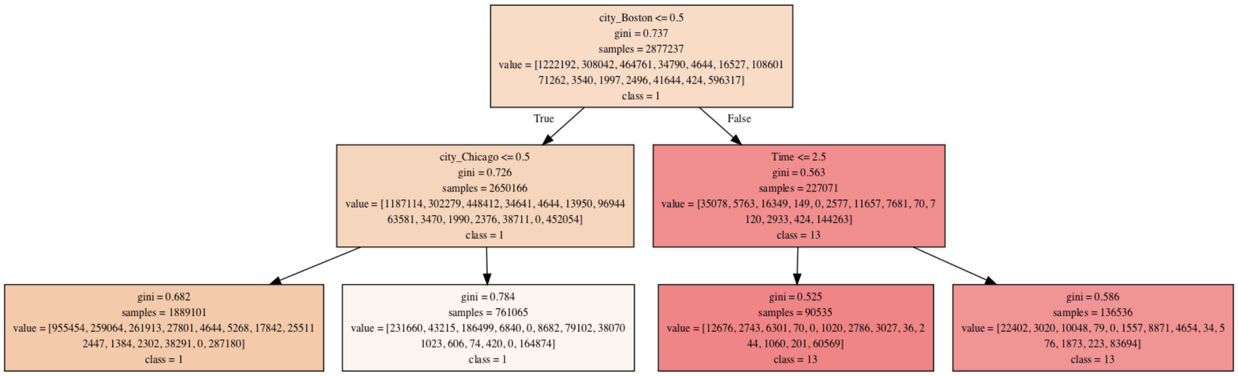
In [362]:
```
dt = tree.DecisionTreeClassifier(max_depth=2)
```

In [363]:
```
dt.fit(X,Y)
```

Out[363]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_dept
h=2,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_stat
e=None,
            splitter='best')
```

```
In [364]: # This code will visualize a decision tree dt,
          # trained with the attributes in X and the class labels in Y
          dt_feature_names = list(X.columns)
          dt_target_names = np.array(Y.unique(),dtype=np.string_)
          result = np.array([s.decode('UTF-8') for s in dt_target_names])
          tree.export_graphviz(dt, out_file='tree.dot',
              feature_names=dt_feature_names, class_names=result,filled=True)
          graph = pydotplus.graph_from_dot_file('tree.dot')
          Image(graph.create_png())
```

Out[364]:



Machine Learning Conclusion :

The decision tree above is split by city first, and this shows that each city has its own criminal patterns. From this information, we performed various analyses on the data based on city and found the insight shown above.

Second Insight: Offense vs Day of the week

Many crimes seem to be committed at a consistent rate over the week or tend to drop off through the weekend. However, it is interesting that homicides seems to increase over the weekend and peak on Sunday. In the US, Sunday is the day people go to church and it seems surprising that this is the day when the most homicides occur.

Creating the copy of the Merge_City

```
In [365]: Merge_C = Merge_City.copy()
In [366]: Merge_C[:2]
```

Out[366]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
1	(40, 50]	2012-10-01 13:00:00	3	aggravated assault	7.0	48.074	scattered clouds	2012.0	Po

Getting weekday from datetime

```
In [367]: Merge_C['WeekDay'] = pd.to_datetime(Merge_C['Datetime']).dt.weekday_na
          me
```

Creating individual dataframes for each crime type in order to compare each crime type over weekday

```
In [368]: Offense_1 = Merge_C[(Merge_C['Offense_Num'] == 1) &\
                                 (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

```
In [369]: Offense_1[:2]
```

Out[369]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
2	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po

```
In [370]: Offense_2 = Merge_C[(Merge_C['Offense_Num'] == 2.0) &\
                                 (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [371]:

```
Offense_2[:2]
```

Out[371]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Yea
18	(40, 50]	2012-10-01 14:00:00	2	burglary	7.0	48.079854	scattered clouds	2012
26	(40, 50]	2012-10-01 16:00:00	2	burglary	7.0	48.110866	scattered clouds	2012

In [372]:

```
Offense_3 = Merge_C[(Merge_C['Offense_Num'] == 3.0) &\
                      (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [373]:

```
Offense_3[:2]
```

Out[373]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year
1	(40, 50]	2012-10-01 13:00:00	3	aggravated assault	7.0	48.074000	scattered clouds	2012.0
6	(40, 50]	2012-10-01 14:00:00	3	aggravated assault	7.0	48.079854	scattered clouds	2012.0

In [374]:

```
Offense_4 = Merge_C[(Merge_C['Offense_Num'] == 4.0) &\
                      (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [375]:

```
Offense_4[:2]
```

Out[375]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Ye
37	(40, 50]	2012-10-01 17:00:00	4	prostitution	7.0	48.126372	scattered clouds	201
179	(40, 50]	2012-10-02 13:00:00	4	rape	7.0	48.436494	scattered clouds	201

In [376]:

```
Offense_5 = Merge_C[(Merge_C['Offense_Num'] == 5.0) &\
                      (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [377]:
```
Offense_5[:2]
```

Out[377]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year
324	(60, 70]	2012-10-03 06:00:00	5	forgery	6.0	63.158	sky is clear	2012.0
325	(60, 70]	2012-10-03 06:00:00	5	forgery	6.0	63.158	sky is clear	2012.0

In [378]:
```
Offense_6 = Merge_C[(Merge_C['Offense_Num'] == 6.0) &\
                            (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [379]:
```
Offense_6[:2]
```

Out[379]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Ye
54	(40, 50]	2012-10-01 18:00:00	6	weapons	7.0	48.141878	scattered clouds	201
274	(70, 80]	2012-10-02 20:00:00	6	weapons	6.0	72.356000	sky is clear	201

In [380]:
```
Offense_7 = Merge_C[(Merge_C['Offense_Num'] == 7.0) &\
                            (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [381]:
```
Offense_7[:2]
```

Out[381]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Yea
55	(40, 50]	2012-10-01 18:00:00	7	drugs	7.0	48.141878	scattered clouds	2012
63	(40, 50]	2012-10-01 19:00:00	7	drugs	7.0	48.157384	scattered clouds	2012

In [382]:
```
Offense_8 = Merge_C[(Merge_C['Offense_Num'] == 8.0) &\
                            (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [383]:

```
Offense_8[:2]
```

Out[383]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Yea
21	(40, 50]	2012-10-01 14:00:00	8	fraud	7.0	48.079854	scattered clouds	2012
35	(40, 50]	2012-10-01 16:00:00	8	fraud	7.0	48.110866	scattered clouds	2012

In [384]:

```
Offense_9 = Merge_C[(Merge_C['Offense_Num'] == 9.0) &\
                     (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [385]:

```
Offense_9[:2]
```

Out[385]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Ye
104	(40, 50]	2012-10-02 01:00:00	9	arson	7.0	48.250421	scattered clouds	201
106	(40, 50]	2012-10-02 01:00:00	9	arson	7.0	48.250421	scattered clouds	201

In [386]:

```
Offense_10 = Merge_C[(Merge_C['Offense_Num'] == 10.0) &\
                     (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [387]:

```
Offense_10[:2]
```

Out[387]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Yea
7732	(50, 60]	2012-11-13 22:00:00	10	kidnap	7.0	50.072	overcast clouds	2012
10663	(40, 50]	2012-12-01 15:00:00	10	kidnap	3.0	46.256	light rain	2012

In [388]:

```
Offense_11 = Merge_C[(Merge_C['Offense_Num'] == 11.0) &\
                     (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [389]:
```
Offense_11[:2]
```

Out[389]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year
2634	(60, 70]	2012-10-15 20:00:00	11	homicide	6.0	62.978	sky is clear	2012.0
3455	(60, 70]	2012-10-20 01:00:00	11	homicide	7.0	60.458	overcast clouds	2012.0

In [390]:
```
Offense_12 = Merge_C[(Merge_C['Offense_Num'] == 12.0) &\
                            (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [391]:
```
Offense_12[:2]
```

Out[391]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Ye
101	(40, 50]	2012-10-01 17:00:00	12	harassment	6.0	47.525116	sky is clear	201
110	(40, 50]	2012-10-01 17:00:00	12	harassment	6.0	47.525116	sky is clear	201

In [392]:
```
Offense_13 = Merge_C[(Merge_C['Offense_Num'] == 13.0) &\
                            (Merge_C.Bins != nan) & (Merge_C.TD_num != nan)]
```

In [393]:
```
Offense_13[:2]
```

Out[393]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Yea
1057	(60, 70]	2015-06-19 03:00:00	13	operating under the influence	6.0	64.3991	sky is clear	2015
1617	(60, 70]	2015-06-21 04:00:00	13	operating under the influence	3.0	61.6622	moderate rain	2015

In [394]:
```
Offense_14 = Merge_C[(Merge_C['Offense_Num'] == 14.0) &\
                            (Merge_C.Bins != nan) & (Merge_C.TD_num!= nan)]
```

In [395]:

```
Offense_14[:2]
```

Out[395]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Yea
7	(40, 50]	2012-10-01 14:00:00	14	vandalism	7.0	48.079854	scattered clouds	2012
20	(40, 50]	2012-10-01 14:00:00	14	trespass	7.0	48.079854	scattered clouds	2012

Assigning numbers to weekdays

In [396]:

```
def dayofweek(row):
    if row == 'Monday':
        return 1
    elif row == 'Tuesday':
        return 2
    elif row == 'Wednesday':
        return 3
    elif row == 'Thursday':
        return 4
    elif row == 'Friday':
        return 5
    elif row== 'Saturday':
        return 6
    elif row== 'Sunday':
        return 7
```

In [397]:

```
Offense_1.head()
```

Out[397]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
2	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
3	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
4	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
5	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po

Calculating the day rate by grouping by weekday for each offense

day rate = (count per offense per weekday) / total count per offense

In [398]:

```
group1=Offense_1.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [399]:

```
group1.head()
```

Out[399]:

	WeekDay	count
0	Friday	233149
1	Monday	231050
2	Saturday	212742
3	Sunday	196107
4	Thursday	221802

In [400]:

```
total_1 = group1['count'].sum()
```

In [401]:

```
group1['day_rate'] = group1['count'] / total_1
```

In [402]:

```
group1['order'] = group1['WeekDay'].apply(dayofweek)
```

In [403]:

```
group1 = group1.sort_values('order',ascending=True)
```

In [404]:

```
group1.head()
```

Out[404]:

	WeekDay	count	day_rate	order
1	Monday	231050	0.149592	1
5	Tuesday	224822	0.145560	2
6	Wednesday	224858	0.145583	3
4	Thursday	221802	0.143605	4
0	Friday	233149	0.150951	5

In [405]:

```
group2=Offense_2.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [406]:

```
total_2 = group2['count'].sum()

group2['day_rate'] = group2['count'] / total_2

group2.head()
```

Out[406]:

	WeekDay	count	day_rate
0	Friday	61602	0.157081
1	Monday	59501	0.151724
2	Saturday	50170	0.127930
3	Sunday	46729	0.119156
4	Thursday	58028	0.147968

In [407]:

```
group2['order'] = group2['WeekDay'].apply(dayofweek)
```

In [408]:

```
group2 = group2.sort_values('order',ascending=True)
```

In [409]:

```
group2.head()
```

Out[409]:

	WeekDay	count	day_rate	order
1	Monday	59501	0.151724	1
5	Tuesday	58549	0.149296	2
6	Wednesday	57588	0.146846	3
4	Thursday	58028	0.147968	4
0	Friday	61602	0.157081	5

In [410]:

```
group3=Offense_3.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [411]:

```
total_3 = group1['count'].sum()
```

In [412]:

```
group3['day_rate'] = group3['count'] / total_3
```

In [413]:

```
group3.head()
```

Out[413]:

	WeekDay	count	day_rate
0	Friday	85614	0.055430
1	Monday	84045	0.054415
2	Saturday	95578	0.061882
3	Sunday	102232	0.066190
4	Thursday	81714	0.052905

In [414]:

```
group3['order'] = group3['WeekDay'].apply(dayofweek)
```

In [415]:

```
group3 = group3.sort_values('order',ascending=True)
```

In [416]:

```
group3.head()
```

Out[416]:

	WeekDay	count	day_rate	order
1	Monday	84045	0.054415	1
5	Tuesday	81131	0.052528	2
6	Wednesday	82433	0.053371	3
4	Thursday	81714	0.052905	4
0	Friday	85614	0.055430	5

In [417]:

```
group4=Offense_4.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [418]:

```
total_4 = group4['count'].sum()
```

In [419]:

```
group4['day_rate'] = group4['count'] / total_4
```

In [420]:

```
group4.head()
```

Out[420]:

	WeekDay	count	day_rate
0	Friday	6903	0.152236
1	Monday	5787	0.127624
2	Saturday	6306	0.139070
3	Sunday	5830	0.128573
4	Thursday	6757	0.149016

In [421]:

```
group4['order'] = group4['WeekDay'].apply(dayofweek)
```

In [422]:

```
group4 = group4.sort_values('order',ascending=True)
```

In [423]:

```
group4.head()
```

Out[423]:

	WeekDay	count	day_rate	order
1	Monday	5787	0.127624	1
5	Tuesday	6781	0.149546	2
6	Wednesday	6980	0.153934	3
4	Thursday	6757	0.149016	4
0	Friday	6903	0.152236	5

In [424]:

```
group5=Offense_5.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [425]:

```
total_5= group5['count'].sum()

group5['day_rate'] = group5['count'] / total_5

group5.head()
```

Out[425]:

	WeekDay	count	day_rate
0	Friday	974	0.171117
1	Monday	799	0.140372
2	Saturday	531	0.093289
3	Sunday	337	0.059206
4	Thursday	1026	0.180253

In [426]:

```
group5['order'] = group5['WeekDay'].apply(dayofweek)
```

In [427]:

```
group5 = group5.sort_values('order',ascending=True)
```

In [428]:

```
group5.head()
```

Out[428]:

	WeekDay	count	day_rate	order
1	Monday	799	0.140372	1
5	Tuesday	1008	0.177091	2
6	Wednesday	1017	0.178672	3
4	Thursday	1026	0.180253	4
0	Friday	974	0.171117	5

In [429]:

```
group6=Offense_6.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [430]:

```
total_6 = group6['count'].sum()

group6['day_rate'] = group6['count'] / total_6

group6.head()
```

Out[430]:

	WeekDay	count	day_rate
0	Friday	3122	0.143382
1	Monday	3019	0.138652
2	Saturday	3397	0.156012
3	Sunday	3087	0.141775
4	Thursday	3160	0.145127

In [431]:

```
group6['order'] = group6['WeekDay'].apply(dayofweek)
```

In [432]:

```
group6 = group6.sort_values('order',ascending=True)
```

```
In [433]: group6.head()
```

Out[433]:

	WeekDay	count	day_rate	order
1	Monday	3019	0.138652	1
5	Tuesday	2978	0.136769	2
6	Wednesday	3011	0.138284	3
4	Thursday	3160	0.145127	4
0	Friday	3122	0.143382	5

```
In [434]: group7=Offense_7.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
          rename(columns ={'Offense_Num':'count'}).reset_index()
```

```
In [435]: total_7 = group1['count'].sum()

          group7['day_rate'] = group7['count'] / total_7

          group7.head()
```

Out[435]:

	WeekDay	count	day_rate
0	Friday	22960	0.014865
1	Monday	20005	0.012952
2	Saturday	19523	0.012640
3	Sunday	17126	0.011088
4	Thursday	22283	0.014427

```
In [436]: group7['order'] = group7['WeekDay'].apply(dayofweek)
```

```
In [437]: group7 = group7.sort_values('order',ascending=True)
```

In [438]:

```
group7.head()
```

Out[438]:

	WeekDay	count	day_rate	order
1	Monday	20005	0.012952	1
5	Tuesday	21969	0.014224	2
6	Wednesday	22723	0.014712	3
4	Thursday	22283	0.014427	4
0	Friday	22960	0.014865	5

In [439]:

```
group8=Offense_8.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [440]:

```
total_8 = group8['count'].sum()

group8['day_rate'] = group8['count'] / total_8

group8.head()
```

Out[440]:

	WeekDay	count	day_rate
0	Friday	14283	0.161865
1	Monday	14504	0.164370
2	Saturday	9628	0.109112
3	Sunday	7353	0.083330
4	Thursday	13543	0.153479

In [441]:

```
group8['order'] = group8['WeekDay'].apply(dayofweek)
```

In [442]:

```
group8 = group8.sort_values('order',ascending=True)
```

In [443]:

```
group8.head()
```

Out[443]:

	WeekDay	count	day_rate	order
1	Monday	14504	0.164370	1
5	Tuesday	14453	0.163792	2
6	Wednesday	14476	0.164053	3
4	Thursday	13543	0.153479	4
0	Friday	14283	0.161865	5

In [444]:

```
group9=Offense_9.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [445]:

```
total_9 = group9['count'].sum()

group9['day_rate'] = group9['count'] / total_9

group9.head()
```

Out[445]:

	WeekDay	count	day_rate
0	Friday	660	0.142641
1	Monday	680	0.146963
2	Saturday	687	0.148476
3	Sunday	698	0.150854
4	Thursday	610	0.131835

In [446]:

```
group9['order'] = group9['WeekDay'].apply(dayofweek)
```

In [447]:

```
group9 = group9.sort_values('order',ascending=True)
```

In [448]:

```
group9.head()
```

Out[448]:

	WeekDay	count	day_rate	order
1	Monday	680	0.146963	1
5	Tuesday	638	0.137886	2
6	Wednesday	654	0.141344	3
4	Thursday	610	0.131835	4
0	Friday	660	0.142641	5

In [449]:

```
group10=Offense_10.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [450]:

```
total_10 = group10['count'].sum()

group10['day_rate'] = group10['count'] / total_10

group10.head()
```

Out[450]:

	WeekDay	count	day_rate
0	Friday	427	0.163164
1	Monday	375	0.143294
2	Saturday	382	0.145969
3	Sunday	347	0.132595
4	Thursday	368	0.140619

In [451]:

```
group10['order'] = group10['WeekDay'].apply(dayofweek)
```

In [452]:

```
group10 = group10.sort_values('order',ascending=True)
```

In [453]:

```
group10.head()
```

Out[453]:

	WeekDay	count	day_rate	order
1	Monday	375	0.143294	1
5	Tuesday	355	0.135652	2
6	Wednesday	363	0.138708	3
4	Thursday	368	0.140619	4
0	Friday	427	0.163164	5

In [454]:

```
group11=Offense_11.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [455]:

```
total_11 = group11['count'].sum()

group11['day_rate'] = group11['count'] / total_11

group11.head()
```

Out[455]:

	WeekDay	count	day_rate
0	Friday	385	0.120993
1	Monday	417	0.131050
2	Saturday	532	0.167190
3	Sunday	581	0.182590
4	Thursday	399	0.125393

In [456]:

```
group11['order'] = group11['WeekDay'].apply(dayofweek)
```

In [457]:

```
group11 = group11.sort_values('order',ascending=True)
```

In [458]:

```
group11.head()
```

Out[458]:

	WeekDay	count	day_rate	order
1	Monday	417	0.131050	1
5	Tuesday	448	0.140792	2
6	Wednesday	420	0.131992	3
4	Thursday	399	0.125393	4
0	Friday	385	0.120993	5

In [459]:

```
group12=Offense_12.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [460]:

```
total_12 = group12['count'].sum()

group12['day_rate'] = group12['count'] / total_12

group12.head()
```

Out[460]:

	WeekDay	count	day_rate
0	Friday	7869	0.148707
1	Monday	8065	0.152411
2	Saturday	6709	0.126786
3	Sunday	6724	0.127069
4	Thursday	7703	0.145570

In [461]:

```
group12['order'] = group12['WeekDay'].apply(dayofweek)
```

In [462]:

```
group12 = group12.sort_values('order',ascending=True)
```

In [463]:

```
group12.head()
```

Out[463]:

	WeekDay	count	day_rate	order
1	Monday	8065	0.152411	1
5	Tuesday	7835	0.148065	2
6	Wednesday	8011	0.151391	3
4	Thursday	7703	0.145570	4
0	Friday	7869	0.148707	5

In [464]:

```
group13=Offense_13.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [465]:

```
total_13 = group13['count'].sum()
group13['day_rate'] = group13['count'] / total_13
group13.head()
```

Out[465]:

	WeekDay	count	day_rate
0	Friday	69	0.146186
1	Monday	53	0.112288
2	Saturday	100	0.211864
3	Sunday	105	0.222458
4	Thursday	53	0.112288

In [466]:

```
group13['order'] = group13['WeekDay'].apply(dayofweek)

group13 = group13.sort_values('order',ascending=True)

group13.head()
```

Out[466]:

	WeekDay	count	day_rate	order
1	Monday	53	0.112288	1
5	Tuesday	41	0.086864	2
6	Wednesday	51	0.108051	3
4	Thursday	53	0.112288	4
0	Friday	69	0.146186	5

In [467]:

```
group14=Offense_14.groupby(['WeekDay']).agg({'Offense_Num':'count'}).\
rename(columns ={'Offense_Num':'count'}).reset_index()
```

In [468]:

```
total_14 = group14['count'].sum()

group14['day_rate'] = group14['count'] / total_14

group14.head()
```

Out[468]:

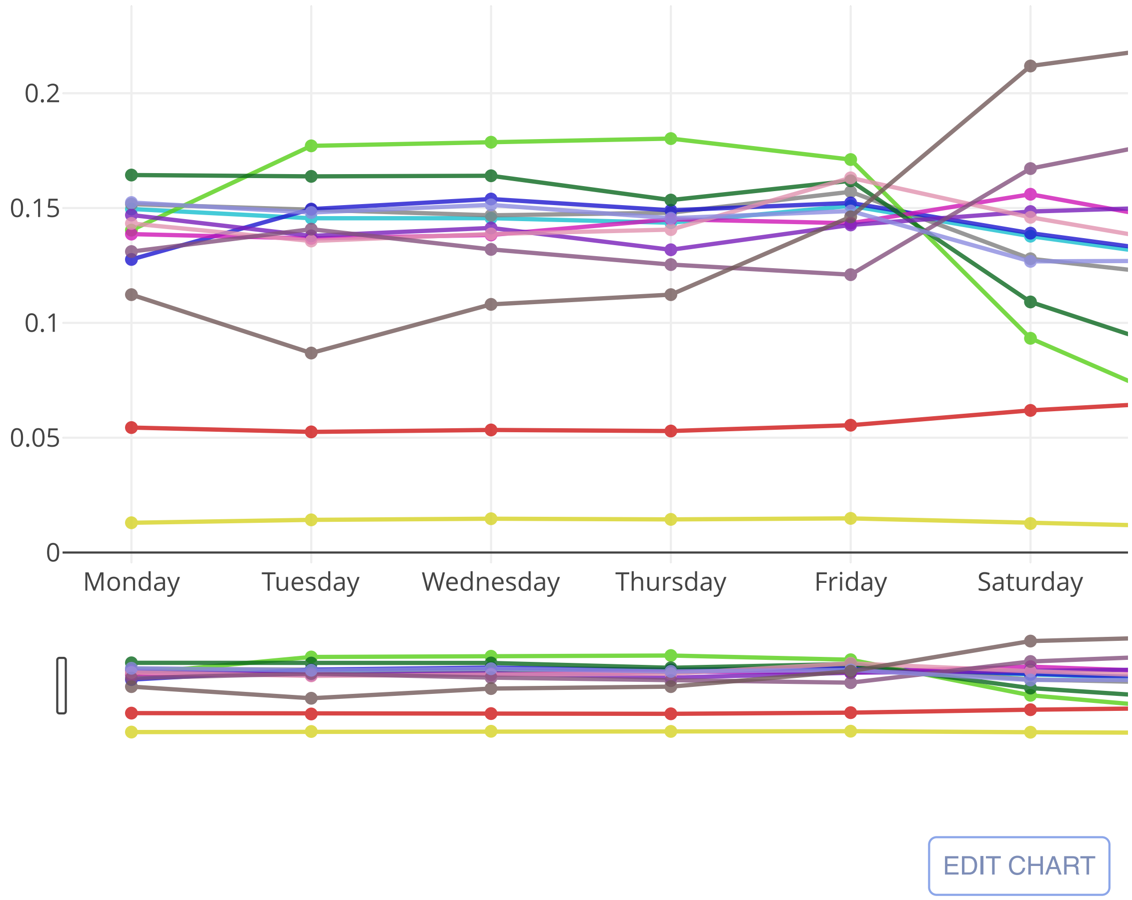
	WeekDay	count	day_rate
0	Friday	113511	0.150932
1	Monday	105781	0.140654
2	Saturday	109158	0.145144
3	Sunday	104611	0.139098
4	Thursday	106279	0.141316

Graph: Crime Rate per day for various Crime Types

In [469]:

```
import plotly.plotly as py
import plotly.graph_objs as go

import pandas as pd

trace_1 = go.Scatter(
    x=group1['WeekDay'],
    y=group1['day_rate'],
    name = "Theft",
    line = dict(color = '#17BECF'),
    opacity = 0.8)
trace_2 = go.Scatter(
    x=group2['WeekDay'],
    y=group2['day_rate'],
    name = "Burglary",
    line = dict(color = '#7F7F7F'),
    opacity = 0.8)
trace_3 = go.Scatter(
    x=group3['WeekDay'],
    y=group3['day_rate'],
    name = "Assault",
    line = dict(color = '#cf1717'),
    opacity = 0.8)
trace_4 = go.Scatter(
    x=group4['WeekDay'],
    y=group4['day_rate'],
    name = "Sex_Crimes",
    line = dict(color = '#1d17cf'),
    opacity = 0.8)
trace_5 = go.Scatter(
    x=group5['WeekDay'],
    y=group5['day_rate'],
    name = "Forgery",
    line = dict(color = '#57cf17'),
    opacity = 0.8)
trace_6 = go.Scatter(
    x=group6['WeekDay'],
    y=group6['day_rate'],
    name = "Weapons",
    line = dict(color = '#cf17b6'),
    opacity = 0.8)
trace_7 = go.Scatter(
    x=group7['WeekDay'],
    y=group7['day_rate'],
    name = "Drug",
    line = dict(color = '#d6d324'),
    opacity = 0.8)
trace_8 = go.Scatter(
    x=group8['WeekDay'],
    y=group8['day_rate'],
    name = "Fraud",
    line = dict(color = '#0b681f'),
    opacity = 0.8)
trace_9 = go.Scatter(
    x=group9['WeekDay'],
    y=group9['day_rate'],
    name = "Arson",
    line = dict(color = '#791fba'),
    opacity = 0.8)
trace_10 = go.Scatter(
    x=group10['WeekDay'],
    y=group10['day_rate'],
    name = "Kidnapping",
    line = dict(color = '#e293b0'),
    opacity = 0.8)
trace_11 = go.Scatter(
    x=group11['WeekDay'],
    y=group11['day_rate'],
    name = "Homicide",
    line = dict(color = '#84507d'),
    opacity = 0.8)
trace_12 = go.Scatter(
    x=group12['WeekDay'],
    y=group12['day_rate'],
    name = "Harass",
    line = dict(color = '#8d8de0'),
    opacity = 0.8)
trace_13 = go.Scatter(
    x=group13['WeekDay'],
    y=group13['day_rate'],
    name = "Driving",
    line = dict(color = '#725959'),
    opacity = 0.8)
data = [trace_1,trace_2,trace_3,trace_4,trace_5,trace_6,trace_7,trace_
8,trace_9,trace_10,trace_11,trace_12,trace_13]


layout = dict(
    title='Crime Rate per day for various Crime Types',
    xaxis= dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1,
                     label='1m',
                     step=group13['WeekDay'],
                     stepmode='backward'),
                dict(count=6,
                     label='6m',
                     step=group13['WeekDay'],
                     stepmode='backward'),
                dict(step='all')

            ])
        ),
        rangeslider=dict(),
        type='weekday'
    )
)

fig = dict(data=data, layout=layout)
py.iplot(fig, filename = "Crime Rate per day for various Crime Types")
```

Out[469]:



Crime Rate per day for various Crime Types

Managerial Insight:

This analysis can be used to promote the training and depoloyment of specialized forces depending on the day of the week. For example, over the weekend, there is a higher necessity for highway patrollers skilled in identifying erratic drivers as well as agents skilled in diffusing situations that could lead to homicide. This prevents unnecessary loss of life on both the victim's, offender's, and law enforcement's side.

Machine Learning:

Random Forest Classifier

In [470]:
```
Merge_City.head(2)
```

Out[470]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
1	(40, 50]	2012-10-01 13:00:00	3	aggravated assault	7.0	48.074	scattered clouds	2012.0	Po

In [471]:
```
Merge_City_ml = Merge_City.copy()
```

In [472]:
```
Merge_City_ml= Merge_City_ml.dropna(how='any',axis=0)
```

Getting date from datetime

In [473]:
```
Merge_City_ml ['Date'] = pd.to_datetime(Merge_City_ml ['Datetime']).dt
.date
```

Getting weekday from datetime

In [474]:
```
Merge_City_ml ['WeekDay'] = pd.to_datetime(Merge_City_ml ['Datetime'])
.dt.weekday
```

Getting time from datetime

In [475]:
```
Merge_City_ml ['Time'] = pd.to_datetime(Merge_City_ml ['Datetime']).dt
.time
```

In [476]:
```
Merge_City_ml.head()
```

Out[476]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
1	(40, 50]	2012-10-01 13:00:00	3	aggravated assault	7.0	48.074	scattered clouds	2012.0	Po
2	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
3	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
4	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po

Assigning number to temperatures for every 20 degrees

In [477]:
```
def tempseries(row):
    if (row >= -20) & (row < 0):
        return 1
    elif (row >= 0) & (row < 21):
        return 2
    elif (row >= 21) & (row < 40):
        return 3
    elif (row >= 40) & (row < 60):
        return 4
    elif (row >= 60) & (row < 80):
        return 5
    elif (row >= 80) & (row < 100):
        return 6
    elif (row >= 100) & (row < 120):
        return 7
```

In [478]:
```
Merge_City_ml['Temp']= Merge_City_ml.Temp.apply(tempseries)
```

In [479]:
```
Merge_City_ml.drop(['Offense_Type','Temp_Des','Datetime','city','Year'
,'Bins'],axis=1,inplace=True)
```

In [480]:

```
Merge_City_ml.head()
```

Out[480]:

	Offense_Num	TD_num	Temp	Date	WeekDay	Time
0	1	7.0	4	2012-10-01	0	13:00:00
1	3	7.0	4	2012-10-01	0	13:00:00
2	1	7.0	4	2012-10-01	0	13:00:00
3	1	7.0	4	2012-10-01	0	13:00:00
4	1	7.0	4	2012-10-01	0	13:00:00

Changing the format of date and time

In [481]:

```
Merge_City_ml['Time']=pd.to_datetime(Merge_City_ml['Time'],format = '%
H:%M:%S' )
```

In [482]:

```
Merge_City_ml['Date']=pd.to_datetime(Merge_City_ml['Date'],format = '%
Y-%m-%d' )
```

Assigning numbers for time with an interval of 6 hours

In [483]:

```
def timeseries(row):
    if (row >= '01') & (row < '07'):
        return 1
    elif (row >= '07') & (row < '13'):
        return 2
    elif (row >= '13') & (row < '19'):
        return 3
    elif (row >= '19') & (row < '25'):
        return 4
```

In [484]:

```
Merge_City_ml['Time']=pd.to_datetime(Merge_City_ml['Time']).dt.strftim
e('%H')
```

In [485]:

```
Merge_City_ml['Month'] = pd.to_datetime(Merge_City_ml['Date']).dt.mont
h
```

In [486]:

```
Merge_City_ml['Day'] = pd.to_datetime(Merge_City_ml['Date']).dt.day
```

In [487]:

```
Merge_City_ml['Time'] = Merge_City_ml['Time'].replace('00','24')
```

In [488]:

```
Merge_City_ml['Time']= Merge_City_ml.Time.apply(timeseries)
```

In [489]:
```
Merge_City_ml.drop(['Date'],axis=1,inplace=True)
```

Grouping crimes into violent, property and other:

1 -- Property Crimes : Theft, Burglary & Arson

2 -- Violent Crimes : Assault , Sex Crimes , Kidnapping , Homicide

3 -- Other Crimes : Forgery , Weapons, Drugs, Fraud,Harass, Driving

In [490]:
```
def crime_cat(row):
    if (row == 1) | (row == 2) | (row ==9):
        return 1
    elif (row == 3) | (row == 4) | (row ==10) | (row == 11):
        return 2
    elif (row == 5) | (row == 6) | (row ==  7) | (row ==12) |(row == 1
3) | (row == 8):
        return 3
```

In [491]:
```
Merge_City_ml.head()
```

Out[491]:

	Offense_Num	TD_num	Temp	WeekDay	Time	Month	Day
0	1	7.0	4	0	3	10	1
1	3	7.0	4	0	3	10	1
2	1	7.0	4	0	3	10	1
3	1	7.0	4	0	3	10	1
4	1	7.0	4	0	3	10	1

In [492]:
```
Merge_City_ml['crime_cat'] = Merge_City_ml['Offense_Num'].apply(crime_
cat)
```

In [493]:
```
Merge_City_dt = Merge_City_ml.copy()
```

In [494]:
```
Merge_City_ml.drop(['Offense_Num'],axis=1,inplace=True)
```

In [495]:
```
Merge_City_ml = Merge_City_ml.dropna(how='any',axis=0)
```

In [496]:
```
X = Merge_City_ml.drop('crime_cat',axis=1)
```

In [497]:
```
Y = Merge_City_ml.crime_cat
```

```
In [498]: from sklearn.model_selection import train_test_split
```

```
In [499]: train_test_split(X,Y,test_size=0.3,random_state = 0)
```

```
Out[499]: [        TD_num  Temp  WeekDay  Time  Month  Day
           858143     7.0     5        5     4     12   10
           643736     7.0     4        2     2     12   23
           243919     3.0     5        6     2      3    5
           11402      6.0     3        0     2     11   12
           119230     7.0     5        1     1      9    9
           77487      5.0     5        3     3      8   20
           56009      5.0     4        0     1      1   14
           47095      7.0     3        1     3      3   19
           534843     6.0     4        2     3      6   24
           223868     6.0     5        3     3     11   28
           110480     7.0     3        5     4      2    9
           798683     6.0     5        5     1      9    3
           119743     6.0     4        3     3      9   11
           991468     5.0     5        6     3      7   16
           84598      5.0     5        3     4      9   10
           613592     6.0     4        2     2     10   29
           187881     7.0     6        5     4      7   23
           98121      5.0     5        1     2     11    3
           118607     6.0     5        2     2      5   15
           391224     6.0     5        6     3     10   12
           13467      7.0     4        1     3     10   16
           148619     6.0     4        5     3      3   30
           421740     7.0     3        5     2      3    1
           251537     6.0     5        2     4      3   29
           105313     7.0     5        4     4     11   13
           998381     6.0     5        3     2      7   27
           358834     6.0     4        5     2      9   10
           93879      7.0     4        4     1      3   29
           237157     7.0     5        0     2      7    8
           132676     3.0     4        1     3     11   25
           ...        ...   ...      ...   ...    ...  ...
           266607     6.0     5        4     3      8    9
           962975     5.0     5        2     3      5   31
           793983     5.0     5        4     4      8   26
           398212     7.0     4        5     3     10   25
           442668     6.0     3        2     2      1   14
           352319     6.0     5        4     2      8   26
           132408     6.0     4        2     4      2    3
           346648     6.0     5        3     2      7   24
           87308      6.0     5        4     1      7    3
           774355     5.0     5        1     2      7   26
           307951     3.0     5        5     4      6    4
           59766      3.0     5        1     1      8   12
           163023     7.0     4        1     2      4    8
           120732     6.0     5        6     4      5   19
           513129     7.0     5        4     2      6   27
           149392     6.0     5        3     3      9    7
           75775      6.0     4        4     1      2   22
```

```
 531395     7.0     5        4     4      7   18
 70514      6.0     3        1     2     12    9
 18245      5.0     4        5     2     11    3
 135298     6.0     4        0     4     12   16
 144805     7.0     5        1     2      3   15
 217787     6.0     4        1     3      2   24
 162735     6.0     3        4     4      2   10
 710421     7.0     3        3     3      3   26
 749401     7.0     4        2     3      5   20
 171867     7.0     5        3     2      8   22
 704385     7.0     3        1     3      3   17
 866529     5.0     4        4     2     12   23
 128986     6.0     3        1     3     11   26

 [1596644 rows x 6 columns],         TD_num  Temp  WeekDay  Time  Mo
nth  Day
 243086     6.0     4        6     1      1    5
 28735      7.0     5        4     4     11   23
 815634     6.0     5        5     2     10    1
 85538      6.0     5        2     1      3   13
 237628     7.0     5        0     4      6   22
 754560     3.0     5        2     3      5   27
 663993     6.0     3        1     3      1   26
 848015     6.0     5        2     4     11   23
 611855     7.0     4        0     1     10   27
 187227     5.0     5        3     2      7   21
 121068     5.0     5        1     4      9    6
 217149     3.0     5        6     3      9   18
 309969     6.0     5        6     1      5   18
 255322     7.0     4        3     2      1   30
 31298      5.0     3        2     4      1   23
 44061      6.0     4        6     3      6   23
 54201      7.0     5        3     3      6   12
 607580     7.0     5        3     3     10   22
 51521      6.0     5        4     4      1    4
 441636     7.0     3        5     2      3   29
 316952     7.0     5        4     4     10    4
 696012     7.0     5        6     4      3   20
 568999     5.0     5        3     3      8   20
 102715     3.0     4        3     2     12   31
 61594      5.0     5        0     2      7    6
 511721     7.0     5        2     3      6   25
 189193     7.0     4        0     2      3   21
 243061     6.0     5        6     4      1    5
 642148     3.0     3        1     2     12    9
 165893     3.0     4        2     1      4   23
 ...        ...   ...      ...   ...    ...  ...
 519431     7.0     5        4     3      7    4
 625306     7.0     3        4     2     11   14
 42930      3.0     4        4     4     11   20
 385067     5.0     2        3     3      1    2
 327888     6.0     4        4     3     10   18
 301423     7.0     4        5     2      5   21
```

```
88779      6.0     4        2     4       3   12
28700      7.0     4        5     3       3   23
729334     7.0     5        5     4       5   14
91154      6.0     5        6     3       8   16
83648      3.0     5        6     4       5   24
319463     6.0     5        2     3       6    4
18573      3.0     4        2     4       2   25
276561     7.0     4        4     3       3   14
292802     6.0     5        5     1       9    7
262715     7.0     4        3     4      11   12
66588      5.0     5        0     2       7   20
53707      7.0     5        3     3       8   15
251977     5.0     4        3     3       1   23
131503     6.0     3        2     3      12    4
301494     6.0     4        1     2       9   17
91582      3.0     4        4     3       3   28
606714     6.0     4        0     3      10   20
676928     6.0     4        2     2       2   17
246422     5.0     4        2     3       3   15
436169     7.0     4        4     4       3   21
36294      7.0     4        2     3       5    8
17944      7.0     4        3     3      12    6
772390     5.0     4        2     2       5    4
201991     6.0     5        3     2      10   17

[684276 rows x 6 columns], 858143    1.0
643736    1.0
243919    1.0
11402     1.0
119230    3.0
77487     1.0
56009     2.0
47095     3.0
534843    1.0
223868    1.0
110480    2.0
798683    2.0
119743    3.0
991468    2.0
84598     1.0
613592    1.0
187881    1.0
98121     1.0
118607    3.0
391224    1.0
13467     3.0
148619    1.0
421740    3.0
251537    1.0
105313    1.0
998381    1.0
358834    3.0
93879     2.0
```

```
237157    1.0
132676    1.0
         ...
266607    1.0
962975    2.0
793983    1.0
398212    1.0
442668    2.0
352319    1.0
132408    1.0
346648    1.0
87308     1.0
774355    1.0
307951    2.0
59766     1.0
163023    1.0
120732    1.0
513129    3.0
149392    1.0
75775     1.0
531395    1.0
70514     1.0
18245     2.0
135298    1.0
144805    1.0
217787    1.0
162735    2.0
710421    2.0
749401    1.0
171867    1.0
704385    1.0
866529    1.0
128986    3.0
Name: crime_cat, Length: 1596644, dtype: float64, 243086    1.0
28735     1.0
815634    1.0
85538     2.0
237628    1.0
754560    3.0
663993    1.0
848015    1.0
611855    2.0
187227    1.0
121068    3.0
217149    1.0
309969    1.0
255322    1.0
31298     1.0
44061     1.0
54201     1.0
607580    2.0
51521     2.0
441636    2.0
```

```
316952    2.0
696012    2.0
568999    1.0
102715    1.0
61594     1.0
511721    3.0
189193    1.0
243061    2.0
642148    3.0
165893    1.0
         ...
519431    3.0
625306    2.0
42930     1.0
385067    2.0
327888    1.0
301423    1.0
88779     1.0
28700     1.0
729334    1.0
91154     1.0
83648     1.0
319463    2.0
18573     2.0
276561    1.0
292802    1.0
262715    1.0
66588     1.0
53707     1.0
251977    1.0
131503    1.0
301494    1.0
91582     3.0
606714    3.0
676928    2.0
246422    1.0
436169    2.0
36294     3.0
17944     1.0
772390    2.0
201991    1.0
Name: crime_cat, Length: 684276, dtype: float64]
```

This method train_test_split() return 4 things :

```
In [500]: X_train, X_test, Y_train, Y_test = \
          train_test_split(X,Y,test_size=0.3,random_state = 0)
```

```
In [501]: len(X)

Out[501]: 2280920
```

```
In [502]: type(X)

Out[502]: pandas.core.frame.DataFrame

In [503]: type(X_train)

Out[503]: pandas.core.frame.DataFrame

In [504]: X_train.head()

Out[504]:
```

	TD_num	Temp	WeekDay	Time	Month	Day
858143	7.0	5	5	4	12	10
643736	7.0	4	2	2	12	23
243919	3.0	5	6	2	3	5
11402	6.0	3	0	2	11	12
119230	7.0	5	1	1	9	9

```
In [505]: X_test.head()

Out[505]:
```

	TD_num	Temp	WeekDay	Time	Month	Day
243086	6.0	4	6	1	1	5
28735	7.0	5	4	4	11	23
815634	6.0	5	5	2	10	1
85538	6.0	5	2	1	3	13
237628	7.0	5	0	4	6	22

```
In [506]: len(X_train)

Out[506]: 1596644

In [507]: len(Y_train)

Out[507]: 1596644

In [508]: len(X_train) / float(len(X))

Out[508]: 0.7

In [509]: len(X_test)

Out[509]: 684276
```

```
In [510]: len(X_train) + len(X_test)

Out[510]: 2280920
```

Train on the trainining set

```
In [511]: from sklearn.ensemble import RandomForestClassifier
```

```
In [512]: cl = RandomForestClassifier(random_state = 0)
```

```
In [513]: cl.fit(X_train,Y_train)

Out[513]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion=
          'gini',
                      max_depth=None, max_features='auto', max_leaf_nodes=None
          ,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
                      oob_score=False, random_state=0, verbose=0, warm_start=F
          alse)
```

```
In [514]: len(X_train)

Out[514]: 1596644
```

```
In [515]: len(X_test)

Out[515]: 684276
```

predict on the test set

```
In [516]: cl.predict(X_test)

Out[516]: array([1., 1., 1., ..., 1., 1., 1.])
```

```
In [517]: (cl.predict(X_test) == Y_test).mean()

Out[517]: 0.6681616774517884
```

Predict on the test set

```
In [518]: y_pred = cl.predict(X_test)
In [519]: y_pred[:20]
```

```
Out[519]: array([1., 1., 1., 1., 1., 1., 1., 1., 2., 1., 1., 1., 2., 1., 1., 1
          ., 1.,
                 1., 1., 1.])
```

Every classifier will also give a prediction output with probability

```
In [520]: cl.predict_proba(X_test)
```

```
Out[520]: array([[0.62818507, 0.28777759, 0.08403735],
                 [0.52970784, 0.28045478, 0.18983738],
                 [0.78577775, 0.1923034 , 0.02191885],
                 ...,
                 [0.73888687, 0.12540546, 0.13570767],
                 [0.62984586, 0.21299841, 0.15715572],
                 [0.85475484, 0.14524516, 0.        ]])
```

```
In [521]: cl.predict_proba(X_test)[1:10,:]
```

```
Out[521]: array([[0.52970784, 0.28045478, 0.18983738],
                 [0.78577775, 0.1923034 , 0.02191885],
                 [0.5225509 , 0.44251912, 0.03492997],
                 [0.71845488, 0.17617855, 0.10536658],
                 [0.60940578, 0.27874562, 0.1118486 ],
                 [0.62571401, 0.37428599, 0.        ],
                 [0.7540502 , 0.20497528, 0.04097452],
                 [0.40780381, 0.47068657, 0.12150962],
                 [0.90874368, 0.09125632, 0.        ]])
```

```
In [522]: y_pred_proba = cl.predict_proba(X_test)[:,1]
```

```
In [523]: y_pred_proba[:20]
```

```
Out[523]: array([0.28777759, 0.28045478, 0.1923034 , 0.44251912, 0.17617855,
                 0.27874562, 0.37428599, 0.20497528, 0.47068657, 0.09125632,
                 0.13246808, 0.22007281, 0.45142857, 0.15581088, 0.02957295,
                 0.25175874, 0.11331148, 0.24236982, 0.25163827, 0.21964148])
```

```
In [524]: X_test.head()
```

Out[524]:

	TD_num	Temp	WeekDay	Time	Month	Day
243086	6.0	4	6	1	1	5
28735	7.0	5	4	4	11	23
815634	6.0	5	5	2	10	1
85538	6.0	5	2	1	3	13
237628	7.0	5	0	4	6	22

Importance of features

```
In [525]: list(zip(X_train,cl.feature_importances_))
```

```
Out[525]: [('TD_num', 0.06631099363095994),
           ('Temp', 0.09327192847572523),
           ('WeekDay', 0.1234815297753725),
           ('Time', 0.09365339238682373),
           ('Month', 0.16252949435267788),
           ('Day', 0.46075266137844073)]
```

Machine Learning Conclusion:

Using the Random Forest Classifier as a prediction model, it can be concluded that the day of the week is the strongest factor in predicting what types of crime will occur given all the data used in this project.We e xplored the relationship between day of the week and crime as shown above.

Third Insight: Temperature Spike vs Crime Spike

It is is apparent that when there is a large spike in temperature over a three day period, there is a corresponding jump in the number of assaults committed. For a large fall in temperature, assaults drop as well.

Boston and Chicago seem to show a similar relation between the size of the temperature spike and the increase in total crimes.

```
In [526]: Merge_City.head()
```

Out[526]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year	
0	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
1	(40, 50]	2012-10-01 13:00:00	3	aggravated assault	7.0	48.074	scattered clouds	2012.0	Po
2	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
3	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po
4	(40, 50]	2012-10-01 13:00:00	1	larceny	7.0	48.074	scattered clouds	2012.0	Po

```
In [527]: merge_all_offense = Merge_City[(Merge_City.Temp!= nan) & (Merge_City.T
          emp.isna()==False)]
```

```
In [528]: merge_all_offense['Datetime'] = pd.to_datetime(merge_all_offense['Date
          time'])
```

```
In [529]: merge_all_offense["Year"] = merge_all_offense['Datetime'].dt.year
```

```
In [530]: merge_all_offense = merge_all_offense[merge_all_offense.Year.between(2
          012,2017)].\
          sort_values(by=['Offense_Num','Datetime'],ascending=[True,True])
```

```
In [531]: merge_all_offense['Date'] = merge_all_offense['Datetime'].dt.date
```

Performed groupby on date and offense_Num to get crime count and average temperature on that date

```
In [532]: merge_all_offense = merge_all_offense.groupby(['Date', 'Offense_Num'])
          .\
          agg({'Temp':'mean', 'Offense_Num':'count'}).rename(columns={'Offense_N
          um':'Offense_count'})
```

```
In [533]: merge_all_offense = merge_all_offense.reset_index()
In [534]: merge_all_offense.head()
```

Out[534]:

	Date	Offense_Num	Temp	Offense_count
0	2012-10-01	1	56.488792	516
1	2012-10-01	2	58.498376	130
2	2012-10-01	3	56.767782	243
3	2012-10-01	4	60.843079	27
4	2012-10-01	6	52.225473	6

Split date based on Offense_Num and performed a 3-day shift on average temperature and offense_count. Then subtracted the shifted data from original data to find the temperature and crime count differences over the 3-day period

Offense 1 Shifted:

```
In [535]: Offense_1_shifted = merge_all_offense[merge_all_offense['Offense_Num']
          ==1.0]
```

```
In [536]: Offense_1_shifted['temp_shift'] = Offense_1_shifted['Temp'].shift(3)
```

```
In [537]: Offense_1_shifted['Offense_shift'] = Offense_1_shifted['Offense_count'
          ].shift(3)
```

```
In [538]: Offense_1_shifted['temp_diff'] = Offense_1_shifted['Temp'] - Offense_1
          _shifted['temp_shift']
```

```
In [539]: Offense_1_shifted['Offense_diff'] = Offense_1_shifted['Offense_count']
          - Offense_1_shifted['Offense_shift']
```

```
In [540]: Offense_1_shifted = Offense_1_shifted.sort_values('temp_diff', ascendi
          ng=True)
```

In [541]:

```
Offense_1_shifted.head()
```

Out[541]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	tem
5768	2014-01-21	1	37.956993	769	64.555507	692.0	-26.5
22624	2017-11-09	1	47.972117	103	66.581774	124.0	-18.6
22271	2017-10-01	1	60.675145	110	77.396449	95.0	-16.7
1293	2013-01-15	1	31.997512	785	46.354335	674.0	-14.3
1508	2013-02-02	1	34.255408	690	48.434466	739.0	-14.1

Offense 2 Shifted:

In [542]:

```
Offense_2_shifted = merge_all_offense[merge_all_offense['Offense_Num']
==2.0]
```

In [543]:

```
Offense_2_shifted['temp_shift'] = Offense_2_shifted['Temp'].shift(3)
```

In [544]:

```
Offense_2_shifted['Offense_shift'] = Offense_2_shifted['Offense_count'
].shift(3)
```

In [545]:

```
Offense_2_shifted['temp_diff'] = Offense_2_shifted['Temp'] - Offense_2
_shifted['temp_shift']
```

In [546]:

```
Offense_2_shifted['Offense_diff'] = Offense_2_shifted['Offense_count']
- Offense_2_shifted['Offense_shift']
```

In [547]:
```
Offense_2_shifted.head()
```

Out[547]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_di
1	2012-10-01	2	58.498376	130	NaN	NaN	NaN
13	2012-10-02	2	66.135756	187	NaN	NaN	NaN
24	2012-10-03	2	65.629290	207	NaN	NaN	NaN
36	2012-10-04	2	64.263662	195	58.498376	130.0	5.765285
49	2012-10-05	2	62.593766	231	66.135756	187.0	-3.54199

In [548]:
```
Offense_2_shifted = Offense_2_shifted.sort_values('temp_diff', ascendi
ng=True)
```

Offense 3 Shifted:

In [549]:
```
Offense_3_shifted = merge_all_offense[merge_all_offense['Offense_Num']
==3.0]
```

In [550]:
```
Offense_3_shifted['temp_shift'] = Offense_3_shifted['Temp'].shift(3)
```

In [551]:
```
Offense_3_shifted['Offense_shift'] = Offense_3_shifted['Offense_count'
].shift(3)
```

In [552]:
```
Offense_3_shifted['temp_diff'] = Offense_3_shifted['Temp'] - Offense_3
_shifted['temp_shift']
```

In [553]:
```
Offense_3_shifted['Offense_diff'] = Offense_3_shifted['Offense_count']
- Offense_3_shifted['Offense_shift']
```

```
In [554]: Offense_3_shifted.head()
```

Out[554]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_di
2	2012-10-01	3	56.767782	243	NaN	NaN	NaN
14	2012-10-02	3	64.074979	408	NaN	NaN	NaN
25	2012-10-03	3	67.536539	347	NaN	NaN	NaN
37	2012-10-04	3	64.884070	345	56.767782	243.0	8.116287
50	2012-10-05	3	66.803309	349	64.074979	408.0	2.728331

```
In [555]: Offense_3_shifted = Offense_3_shifted.sort_values('temp_diff', ascendi
          ng=True)
```

Offense 4 Shifted:

```
In [556]: Offense_4_shifted = merge_all_offense[merge_all_offense['Offense_Num']
          ==4.0]
```

```
In [557]: Offense_4_shifted['temp_shift'] = Offense_4_shifted['Temp'].shift(3)
```

```
In [558]: Offense_4_shifted['Offense_shift'] = Offense_4_shifted['Offense_count'
          ].shift(3)
```

```
In [559]: Offense_4_shifted['temp_diff'] = Offense_4_shifted['Temp'] - Offense_4
          _shifted['temp_shift']
```

```
In [560]: Offense_4_shifted['Offense_diff'] = Offense_4_shifted['Offense_count']
          - Offense_4_shifted['Offense_shift']
```

```
In [561]: Offense_4_shifted.head()
```

Out[561]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_di
3	2012-10-01	4	60.843079	27	NaN	NaN	NaN
15	2012-10-02	4	68.775897	24	NaN	NaN	NaN
26	2012-10-03	4	68.663300	20	NaN	NaN	NaN
38	2012-10-04	4	69.711500	24	60.843079	27.0	8.86842
51	2012-10-05	4	63.087452	31	68.775897	24.0	-5.68844

```
In [562]: Offense_4_shifted = Offense_4_shifted.sort_values('temp_diff', ascendi
          ng=True)
```

Offense 5 Shifted:

```
In [563]: Offense_5_shifted = merge_all_offense[merge_all_offense['Offense_Num']
          ==5.0]
```

```
In [564]: Offense_5_shifted['temp_shift'] = Offense_5_shifted['Temp'].shift(3)
```

```
In [565]: Offense_5_shifted['Offense_shift'] = Offense_5_shifted['Offense_count'
          ].shift(3)
```

```
In [566]: Offense_5_shifted['temp_diff'] = Offense_5_shifted['Temp'] - Offense_5
          _shifted['temp_shift']
```

```
In [567]: Offense_5_shifted['Offense_diff'] = Offense_5_shifted['Offense_count']
          - Offense_5_shifted['Offense_shift']
```

In [568]:

```
Offense_5_shifted.head()
```

Out[568]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_d
27	2012-10-03	5	63.158000	2	NaN	NaN	NaN
39	2012-10-04	5	48.525636	11	NaN	NaN	NaN
52	2012-10-05	5	53.474000	6	NaN	NaN	NaN
76	2012-10-07	5	72.230000	1	63.158000	2.0	9.072000
87	2012-10-08	5	59.936000	2	48.525636	11.0	11.41036

In [569]:

```
Offense_5_shifted = Offense_5_shifted.sort_values('temp_diff', ascendi
ng=True)
```

Offense 6 Shifted:

In [570]:

```
Offense_6_shifted = merge_all_offense[merge_all_offense['Offense_Num']
==6.0]
```

In [571]:

```
Offense_6_shifted['temp_shift'] = Offense_6_shifted['Temp'].shift(3)
```

In [572]:

```
Offense_6_shifted['Offense_shift'] = Offense_6_shifted['Offense_count'
].shift(3)
```

In [573]:

```
Offense_6_shifted['temp_diff'] = Offense_6_shifted['Temp'] - Offense_6
_shifted['temp_shift']
```

In [574]:

```
Offense_6_shifted['Offense_diff'] = Offense_6_shifted['Offense_count']
- Offense_6_shifted['Offense_shift']
```

```
In [575]: Offense_6_shifted.head()
```

Out[575]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_d
4	2012-10-01	6	52.225473	6	NaN	NaN	NaN
16	2012-10-02	6	62.116455	17	NaN	NaN	NaN
28	2012-10-03	6	66.146000	9	NaN	NaN	NaN
40	2012-10-04	6	69.573200	5	52.225473	6.0	17.34772
53	2012-10-05	6	67.067273	11	62.116455	17.0	4.950818

```
In [576]: Offense_6_shifted = Offense_6_shifted.sort_values('temp_diff', ascendi
          ng=True)
```

Offense 7 Shifted:

```
In [577]: Offense_7_shifted = merge_all_offense[merge_all_offense['Offense_Num']
          ==7.0]
```

```
In [578]: Offense_7_shifted['temp_shift'] = Offense_7_shifted['Temp'].shift(3)
```

```
In [579]: Offense_7_shifted['Offense_shift'] = Offense_7_shifted['Offense_count'
          ].shift(3)
```

```
In [580]: Offense_7_shifted['temp_diff'] = Offense_7_shifted['Temp'] - Offense_7
          _shifted['temp_shift']
```

```
In [581]: Offense_7_shifted['Offense_diff'] = Offense_7_shifted['Offense_count']
          - Offense_7_shifted['Offense_shift']
```

In [582]:
```
Offense_7_shifted.head()
```

Out[582]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_di
5	2012-10-01	7	51.684050	68	NaN	NaN	NaN
17	2012-10-02	7	59.747168	119	NaN	NaN	NaN
29	2012-10-03	7	61.944973	111	NaN	NaN	NaN
41	2012-10-04	7	61.623046	109	51.684050	68.0	9.938995
54	2012-10-05	7	63.853732	112	59.747168	119.0	4.106564

In [583]:
```
Offense_7_shifted = Offense_7_shifted.sort_values('temp_diff', ascendi
ng=True)
```

Offense 8 Shifted:

In [584]:
```
Offense_8_shifted = merge_all_offense[merge_all_offense['Offense_Num']
==8.0]
```

In [585]:
```
Offense_8_shifted['temp_shift'] = Offense_8_shifted['Temp'].shift(3)
```

In [586]:
```
Offense_8_shifted['Offense_shift'] = Offense_8_shifted['Offense_count'
].shift(3)
```

In [587]:
```
Offense_8_shifted['temp_diff'] = Offense_8_shifted['Temp'] - Offense_8
_shifted['temp_shift']
```

In [588]:
```
Offense_8_shifted['Offense_diff'] = Offense_8_shifted['Offense_count']
- Offense_8_shifted['Offense_shift']
```

```
In [589]: Offense_8_shifted.head()
```

Out[589]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_di
6	2012-10-01	8	50.199380	42	NaN	NaN	NaN
18	2012-10-02	8	57.776437	78	NaN	NaN	NaN
30	2012-10-03	8	57.703870	92	NaN	NaN	NaN
42	2012-10-04	8	57.045714	70	50.199380	42.0	6.84633
55	2012-10-05	8	60.407214	56	57.776437	78.0	2.63077

```
In [590]: Offense_8_shifted = Offense_8_shifted.sort_values('temp_diff', ascendi
          ng=True)
```

Offense 9 Shifted:

```
In [591]: Offense_9_shifted = merge_all_offense[merge_all_offense['Offense_Num']
          ==9.0]
```

```
In [592]: Offense_9_shifted['temp_shift'] = Offense_9_shifted['Temp'].shift(3)
```

```
In [593]: Offense_9_shifted['Offense_shift'] = Offense_9_shifted['Offense_count'
          ].shift(3)
```

```
In [594]: Offense_9_shifted['temp_diff'] = Offense_9_shifted['Temp'] - Offense_9
          _shifted['temp_shift']
```

```
In [595]: Offense_9_shifted['Offense_diff'] = Offense_9_shifted['Offense_count']
          - Offense_9_shifted['Offense_shift']
```

```
In [596]: Offense_9_shifted.head()
```

Out[596]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_d
7	2012-10-01	9	59.611368	2	NaN	NaN	NaN
19	2012-10-02	9	54.175356	5	NaN	NaN	NaN
31	2012-10-03	9	55.778000	1	NaN	NaN	NaN
43	2012-10-04	9	64.004000	1	59.611368	2.0	4.39263
56	2012-10-05	9	74.570000	1	54.175356	5.0	20.3946

```
In [597]: Offense_9_shifted = Offense_9_shifted.sort_values('temp_diff', ascendi
          ng=True)
```

Offense 10 Shifted:

```
In [598]: Offense_10_shifted = merge_all_offense[merge_all_offense['Offense_Num'
          ]==10.0]
```

```
In [599]: Offense_10_shifted['temp_shift'] = Offense_10_shifted['Temp'].shift(3)
```

```
In [600]: Offense_10_shifted['Offense_shift'] = Offense_10_shifted['Offense_coun
          t'].shift(3)
```

```
In [601]: Offense_10_shifted['temp_diff'] = Offense_10_shifted['Temp'] - Offense
          _10_shifted['temp_shift']
```

```
In [602]: Offense_10_shifted['Offense_diff'] = Offense_10_shifted['Offense_count
          '] - Offense_10_shifted['Offense_shift']
```

In [603]:

```
Offense_10_shifted.head()
```

Out[603]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_d
8	2012-10-01	10	52.953833	1	NaN	NaN	NaN
20	2012-10-02	10	62.060000	1	NaN	NaN	NaN
32	2012-10-03	10	75.650000	2	NaN	NaN	NaN
44	2012-10-04	10	64.832000	2	52.953833	1.0	11.87816
68	2012-10-06	10	47.660000	2	62.060000	1.0	-14.4000

In [604]:

```
Offense_10_shifted = Offense_10_shifted.sort_values('temp_diff', ascen
ding=True)
```

Offense 11 shifted:

In [605]:

```
Offense_11_shifted = merge_all_offense[merge_all_offense['Offense_Num'
]==11.0]
```

In [606]:

```
Offense_11_shifted['temp_shift'] = Offense_11_shifted['Temp'].shift(3)
```

In [607]:

```
Offense_11_shifted['Offense_shift'] = Offense_11_shifted['Offense_coun
t'].shift(3)
```

In [608]:

```
Offense_11_shifted['temp_diff'] = Offense_11_shifted['Temp'] - Offense
_11_shifted['temp_shift']
```

In [609]:

```
Offense_11_shifted['Offense_diff'] = Offense_11_shifted['Offense_count
'] - Offense_11_shifted['Offense_shift']
```

In [610]:

```
Offense_11_shifted.head()
```

Out[610]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_di
9	2012-10-01	11	65.625437	1	NaN	NaN	NaN
45	2012-10-04	11	73.436000	1	NaN	NaN	NaN
57	2012-10-05	11	74.093000	2	NaN	NaN	NaN
69	2012-10-06	11	68.288000	3	65.625437	1.0	2.66256
93	2012-10-08	11	76.280000	1	73.436000	1.0	2.84400

In [611]:

```
Offense_11_shifted = Offense_11_shifted.sort_values('temp_diff', ascending=True)
```

Offense 12 Shifted:

In [612]:

```
Offense_12_shifted = merge_all_offense[merge_all_offense['Offense_Num']==12.0]
```

In [613]:

```
Offense_12_shifted['temp_shift'] = Offense_12_shifted['Temp'].shift(3)
```

In [614]:

```
Offense_12_shifted['Offense_shift'] = Offense_12_shifted['Offense_count'].shift(3)
```

In [615]:

```
Offense_12_shifted['temp_diff'] = Offense_12_shifted['Temp'] - Offense_12_shifted['temp_shift']
```

In [616]:

```
Offense_12_shifted['Offense_diff'] = Offense_12_shifted['Offense_count'] - Offense_12_shifted['Offense_shift']
```

In [617]:
```
Offense_12_shifted.head()
```

Out[617]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp_
10	2012-10-01	12	59.504697	20	NaN	NaN	NaN
21	2012-10-02	12	71.317838	31	NaN	NaN	NaN
33	2012-10-03	12	78.857000	18	NaN	NaN	NaN
46	2012-10-04	12	65.195938	32	59.504697	20.0	5.69124
58	2012-10-05	12	61.083950	40	71.317838	31.0	-10.2338

In [618]:
```
Offense_12_shifted = Offense_12_shifted.sort_values('temp_diff', ascen
ding=True)
```

Offense 13 Shifted:

In [619]:
```
Offense_13_shifted = merge_all_offense[merge_all_offense['Offense_Num'
]==13.0]
```

In [620]:
```
Offense_13_shifted['temp_shift'] = Offense_13_shifted['Temp'].shift(3)
```

In [621]:
```
Offense_13_shifted['Offense_shift'] = Offense_13_shifted['Offense_coun
t'].shift(3)
```

In [622]:
```
Offense_13_shifted['temp_diff'] = Offense_13_shifted['Temp'] - Offense
_13_shifted['temp_shift']
```

In [623]:
```
Offense_13_shifted['Offense_diff'] = Offense_13_shifted['Offense_count
'] - Offense_13_shifted['Offense_shift']
```

```
In [624]: Offense_13_shifted.head()
```

Out[624]:

	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	temp
11959	2015-06-19	13	64.399100	1	NaN	NaN	NaN
11984	2015-06-21	13	61.662200	1	NaN	NaN	NaN
11998	2015-06-22	13	66.206600	1	NaN	NaN	NaN
12034	2015-06-25	13	60.648800	1	64.3991	1.0	-3.75
12073	2015-06-28	13	62.114232	1	61.6622	1.0	0.452

```
In [625]: Offense_13_shifted = Offense_13_shifted.sort_values('temp_diff', ascen
          ding=True)
```

Split date based on city and performed a 3-day shift on average temperature and offense_count. Then subtracted the shifted data from original data to find the temperature and crime count differences over the 3-day period

```
In [626]: merge_all_shift = Merge_City.copy()
```

```
In [627]: merge_all_shift['Datetime'] = pd.to_datetime(Merge_City['Datetime'])
```

```
In [628]: merge_all_shift["year"] = merge_all_shift['Datetime'].dt.year
```

```
In [629]: merge_all_shift = merge_all_shift[merge_all_shift.year.between(2012,20
          17)]
```

```
In [630]: merge_all_shift = merge_all_shift[merge_all_shift.Temp.isna()==False]
```

```
In [631]: merge_all_shift['date'] = merge_all_shift['Datetime'].dt.date
```

Houston Shifted:

```
In [632]: Houston_shifted = merge_all_shift[(merge_all_shift['city'] == 'Houston
          ')].sort_values('Datetime',ascending=True)
```

In [633]:
```
Houston_shifted['city'] = 'Houston'
```

In [634]:
```
Houston_shifted.head()
```

Out[634]:

	Bins	Datetime	Offense_Num	Offense_Type	TD_num	Temp	Temp_Des	Year
0	(50, 60]	2012-10-01 22:00:00	3	aggravated assault	7.0	59.795091	scattered clouds	2012.0
1	(70, 80]	2012-10-15 14:00:00	4	rape	7.0	76.118000	broken clouds	2012.0
2	NaN	2012-10-20 22:00:00	1	theft	7.0	82.004000	few clouds	2012.0
3	(60, 70]	2012-11-01 02:00:00	4	rape	7.0	63.662000	scattered clouds	2012.0
4	(70, 80]	2012-11-01 18:00:00	2	burglary	7.0	78.872000	scattered clouds	2012.0

In [635]:
```
Houston_shifted = Houston_shifted.groupby(['date', 'city']).agg({'Temp
':'mean', 'Offense_Num':'count'}).\
rename(columns={'Offense_Num':'Offense_count'})
```

In [636]:
```
Houston_shifted['temp_shift'] = Houston_shifted['Temp'].shift(3)
```

In [637]:
```
Houston_shifted['Offense_shift'] = Houston_shifted['Offense_count'].sh
ift(3)
```

In [638]:
```
Houston_shifted['temp_diff'] = Houston_shifted['Temp'] -Houston_shifte
d['temp_shift']
```

In [639]:
```
Houston_shifted['Offense_diff'] = Houston_shifted['Offense_count'] - H
ouston_shifted['Offense_shift']
```

In [640]:
```
Houston_shifted = Houston_shifted.sort_values('temp_diff', ascending=T
rue)
```

In [641]:

```
Houston_shifted.head()
```

Out[641]:

		Temp	Offense_count	temp_shift	Offense_shift	temp_diff	Offer
date	city						
2012-11-29	Houston	42.566000	1	82.004000	1.0	-39.438000	0.0
2014-11-14	Houston	33.547864	11	69.154850	4.0	-35.606986	7.0
2014-02-27	Houston	36.140000	1	70.898000	1.0	-34.758000	0.0
2015-12-28	Houston	44.237877	333	76.348285	213.0	-32.110409	120.(
2016-12-20	Houston	43.062225	338	74.787350	402.0	-31.725126	-64.(

Boston Shifted:

In [642]:

```
Boston_shifted = merge_all_shift[(merge_all_shift['city'] == 'Boston')
].sort_values('Datetime',ascending=True)
```

In [643]:

```
Boston_shifted['city'] = 'Boston'
```

In [644]:

```
Boston_shifted = Boston_shifted.groupby(['date', 'city']).agg({'Temp':
'mean', 'Offense_Num':'count'}).\
rename(columns={'Offense_Num':'Offense_count'})
```

In [645]:

```
Boston_shifted['temp_shift'] = Boston_shifted['Temp'].shift(3)
```

In [646]:

```
Boston_shifted['Offense_shift'] = Boston_shifted['Offense_count'].shif
t(3)
```

In [647]:

```
Boston_shifted['temp_diff'] = Boston_shifted['Temp'] - Boston_shifted[
'temp_shift']
```

In [648]:

```
Boston_shifted['Offense_diff'] = Boston_shifted['Offense_count'] - Bos
ton_shifted['Offense_shift']
```

In [649]:

```
Boston_shifted = Boston_shifted.sort_values('temp_diff', ascending=Tru
e)
```

In [650]:
```
Boston_shifted.head()
```

Out[650]:

		Temp	Offense_count	temp_shift	Offense_shift	temp_diff	Offens
date	**city**						
2016-04-04	**Boston**	26.612356	242	63.713273	350.0	-37.100917	-108.0
2017-03-04	**Boston**	19.254029	210	52.712857	308.0	-33.458829	-98.0
2017-03-11	**Boston**	18.920469	256	51.133935	277.0	-32.213466	-21.0
2017-03-12	**Boston**	17.911119	218	46.099040	275.0	-28.187921	-57.0
2017-03-05	**Boston**	19.646488	209	47.445343	268.0	-27.798855	-59.0

Chicago Shifted:

In [651]:
```
Chicago_shifted = merge_all_shift[(merge_all_shift['city'] == 'Chicago
')].sort_values('Datetime',ascending=True)
```

In [652]:
```
Chicago_shifted['city'] = 'Chicago'
```

In [653]:
```
Chicago_shifted = Chicago_shifted.groupby(['date', 'city']).agg({'Temp
':'mean', 'Offense_Num':'count'}).\
rename(columns={'Offense_Num':'Offense_count'})
```

In [654]:
```
Chicago_shifted['temp_shift'] = Chicago_shifted['Temp'].shift(3)
```

In [655]:
```
Chicago_shifted['Offense_shift'] = Chicago_shifted['Offense_count'].sh
ift(3)
```

In [656]:
```
Chicago_shifted['temp_diff'] = Chicago_shifted['Temp'] - Chicago_shift
ed['temp_shift']
```

In [657]:
```
Chicago_shifted['Offense_diff'] = Chicago_shifted['Offense_count'] - C
hicago_shifted['Offense_shift']
```

In [658]:
```
Chicago_shifted = Chicago_shifted.sort_values('temp_diff', ascending=T
rue)
```

```
In [659]: Chicago_shifted.head()
```

Out[659]:

		Temp	Offense_count	temp_shift	Offense_shift	temp_diff	Offer
date	city						
2013-02-02	Chicago	7.642741	740	49.976703	855.0	-42.333963	-115.
2014-01-21	Chicago	9.804061	594	51.732674	657.0	-41.928614	-63.0
2016-01-18	Chicago	2.129773	7	39.490382	10.0	-37.360609	-3.0
2016-01-11	Chicago	3.762351	3	38.799339	8.0	-35.036988	-5.0
2013-12-07	Chicago	15.343603	650	49.887032	866.0	-34.543429	-216.

Atlanta Shifted:

```
In [660]: Atlanta_shifted = merge_all_shift[(merge_all_shift['city'] == 'Atlanta
          ')].sort_values('Datetime',ascending=True)
```

```
In [661]: Atlanta_shifted['city'] = 'Atlanta'
```

```
In [662]: Atlanta_shifted = Atlanta_shifted.groupby(['date', 'city']).agg({'Temp
          ':'mean', 'Offense_Num':'count'}).\
          rename(columns={'Offense_Num':'Offense_count'})
```

```
In [663]: Atlanta_shifted['temp_shift'] = Atlanta_shifted['Temp'].shift(3)
```

```
In [664]: Atlanta_shifted['Offense_shift'] = Atlanta_shifted['Offense_count'].sh
          ift(3)
```

```
In [665]: Atlanta_shifted['temp_diff'] = Atlanta_shifted['Temp'] -Atlanta_shifte
          d['temp_shift']
```

```
In [666]: Atlanta_shifted['Offense_diff'] = Atlanta_shifted['Offense_count'] - A
          tlanta_shifted['Offense_shift']
```

```
In [667]: Atlanta_shifted = Atlanta_shifted.sort_values('temp_diff', ascending=T
          rue)
```

```
In [668]: Atlanta_shifted.head()
```

Out[668]:

		Temp	Offense_count	temp_shift	Offense_shift	temp_diff	Offens
date	**city**						
2014-01-22	**Atlanta**	21.872492	61	52.240220	90.0	-30.367728	-29.0
2013-12-25	**Atlanta**	32.388957	46	61.571614	83.0	-29.182658	-37.0
2017-01-07	**Atlanta**	25.921564	55	52.155114	70.0	-26.233551	-15.0
2015-01-07	**Atlanta**	32.972805	57	58.919956	86.0	-25.947151	-29.0
2013-12-09	**Atlanta**	44.607541	95	70.410141	92.0	-25.802600	3.0

Los Angeles Shifted:

```
In [669]: LA_shifted = merge_all_shift[(merge_all_shift['city'] == 'Los Angeles'
          )].\
          sort_values('Datetime',ascending=True)
```

```
In [670]: LA_shifted['city'] = 'Los Angeles'
```

```
In [671]: LA_shifted = LA_shifted.groupby(['date', 'city']).agg({'Temp':'mean',
          'Offense_Num':'count'}).\
          rename(columns={'Offense_Num':'Offense_count'})
```

```
In [672]: LA_shifted['temp_shift'] = LA_shifted['Temp'].shift(3)
```

```
In [673]: LA_shifted['Offense_shift'] = LA_shifted['Offense_count'].shift(3)
```

```
In [674]: LA_shifted['temp_diff'] = LA_shifted['Temp'] - LA_shifted['temp_shift'
          ]
```

```
In [675]: LA_shifted['Offense_diff'] = LA_shifted['Offense_count'] - LA_shifted[
          'Offense_shift']
```

```
In [676]: LA_shifted = LA_shifted.sort_values('temp_diff', ascending=True)
In [677]: LA_shifted.head()
```

Out[677]:

		Temp	Offense_count	temp_shift	Offense_shift	temp_diff	Offen
date	**city**						
2014-01-21	**Los Angeles**	62.349024	492	82.191600	499.0	-19.842576	-7.0
2013-01-29	**Los Angeles**	39.683273	531	59.176386	533.0	-19.493113	-2.0
2013-10-10	**Los Angeles**	51.601575	508	70.352429	549.0	-18.750854	-41.0
2014-01-20	**Los Angeles**	60.450708	506	77.402863	517.0	-16.952155	-11.0
2015-04-07	**Los Angeles**	44.330755	513	60.788547	524.0	-16.457792	-11.0

Portland Shifted:

```
In [678]: Portland_shifted = merge_all_shift[(merge_all_shift['city'] == 'Portland')].\
          sort_values('Datetime',ascending=True)
```

```
In [679]: Portland_shifted['city'] = 'Portland'
```

```
In [680]: Portland_shifted = Portland_shifted.groupby(['date', 'city']).agg({'Temp':'mean', 'Offense_Num':'count'}).\
          rename(columns={'Offense_Num':'Offense_count'})
```

```
In [681]: Portland_shifted['temp_shift'] = Portland_shifted['Temp'].shift(3)
```

```
In [682]: Portland_shifted['Offense_shift'] = Portland_shifted['Offense_count'].shift(3)
```

```
In [683]: Portland_shifted['temp_diff'] = Portland_shifted['Temp'] - Portland_shifted['temp_shift']
```

```
In [684]: Portland_shifted['Offense_diff'] = Portland_shifted['Offense_count'] - Portland_shifted['Offense_shift']
```

```
In [685]: Portland_shifted = Portland_shifted.sort_values('temp_diff', ascending=True)
In [686]: Portland_shifted.head()
```

Out[686]:

		Temp	Offense_count	temp_shift	Offense_shift	temp_diff	Offe
date	city						
2014-11-30	Portland	27.115610	133	53.112179	98.0	-25.996569	35.0
2014-12-01	Portland	26.726891	174	51.759751	160.0	-25.032860	14.0
2014-01-21	Portland	35.339874	175	59.727474	175.0	-24.387600	0.0
2013-12-04	Portland	27.740482	193	50.943837	147.0	-23.203355	46.0
2015-09-15	Portland	55.204408	157	76.968050	160.0	-21.763642	-3.0

Seattle Shifted:

```
In [687]: Seattle_shifted = merge_all_shift[(merge_all_shift['city'] == 'Seattle
          ')].sort_values('Datetime',ascending=True)
```

```
In [688]: Seattle_shifted['city'] = 'Seattle'
```

```
In [689]: Seattle_shifted = Seattle_shifted.groupby(['date', 'city']).agg({'Temp
          ':'mean', 'Offense_Num':'count'}).\
          rename(columns={'Offense_Num':'Offense_count'})
```

```
In [690]: Seattle_shifted['temp_shift'] = Seattle_shifted['Temp'].shift(3)
```

```
In [691]: Seattle_shifted['Offense_shift'] = Seattle_shifted['Offense_count'].sh
          ift(3)
```

```
In [692]: Seattle_shifted['temp_diff'] = Seattle_shifted['Temp'] - Seattle_shift
          ed['temp_shift']
```

```
In [693]: Seattle_shifted['Offense_diff'] = Seattle_shifted['Offense_count'] - S
          eattle_shifted['Offense_shift']
```

```
In [694]: Seattle_shifted = Seattle_shifted.sort_values('temp_diff', ascending=T
          rue)
```

```
In [695]: Seattle_shifted.head()
```

Out[695]:

		Temp	Offense_count	temp_shift	Offense_shift	temp_diff	Offens
date	**city**						
2014-11-30	**Seattle**	26.873927	133	54.623825	108.0	-27.749898	25.0
2014-12-01	**Seattle**	26.065167	174	51.633077	112.0	-25.567910	62.0
2014-01-21	**Seattle**	36.323896	425	57.556192	266.0	-21.232295	159.0
2013-12-04	**Seattle**	28.972966	356	49.633715	295.0	-20.660749	61.0
2015-11-20	**Seattle**	34.157897	196	52.408345	162.0	-18.250449	34.0

Graphs:

1) Crime spikes due to Temperature spikes over 3 day period by Offense Types

2) Crime spikes due to Temperature spikes over 3 day period by city

```
In [696]: shift_merge = pd.concat([Offense_1_shifted, Offense_2_shifted, Offense
          _3_shifted, Offense_4_shifted, \
                                   Offense_5_shifted, Offense_6_shifted, Offense
          _7_shifted, Offense_8_shifted, \
                                   Offense_9_shifted, Offense_10_shifted,Offense
          _11_shifted, Offense_12_shifted, \
                                   Offense_13_shifted], join='outer')
```

```
In [697]: shift_merge = shift_merge.reset_index()
```

In [698]:

```
shift_merge.head()
```

Out[698]:

	index	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	t
0	5768	2014-01-21	1	37.956993	769	64.555507	692.0	-2
1	22624	2017-11-09	1	47.972117	103	66.581774	124.0	-1
2	22271	2017-10-01	1	60.675145	110	77.396449	95.0	-1
3	1293	2013-01-15	1	31.997512	785	46.354335	674.0	-1
4	1508	2013-02-02	1	34.255408	690	48.434466	739.0	-1

In [699]:

```
shift_merge_cities = pd.concat([Houston_shifted, Boston_shifted, Chica
go_shifted,\
                                Portland_shifted, Seattle_shifted, LA_
shifted,\
                                Atlanta_shifted], join='outer')
```

In [700]:

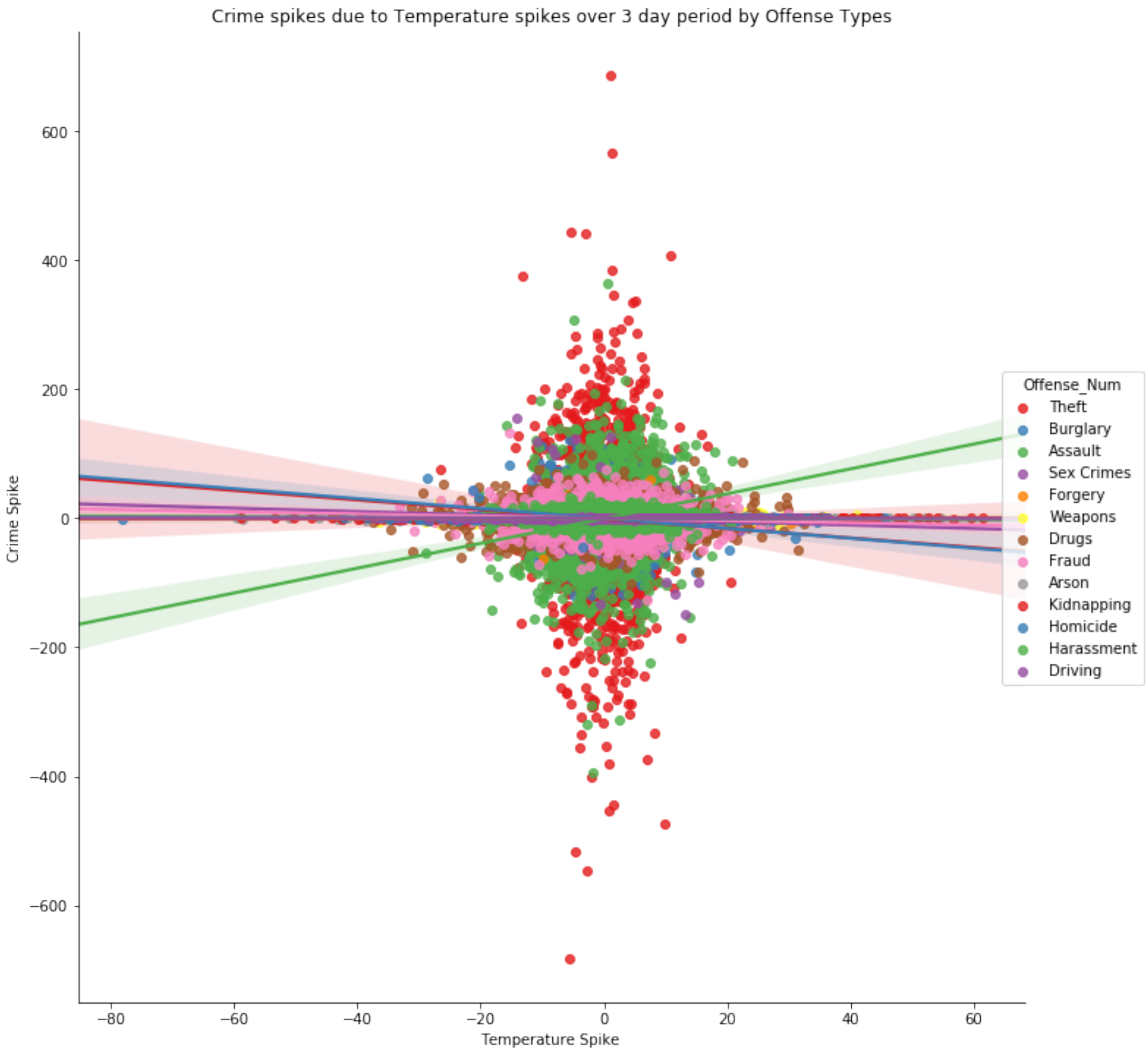
```
shift_merge_cities = shift_merge_cities.reset_index()
```

Temperature spikes vs. Offense spikes per Offense Type:

In [701]:

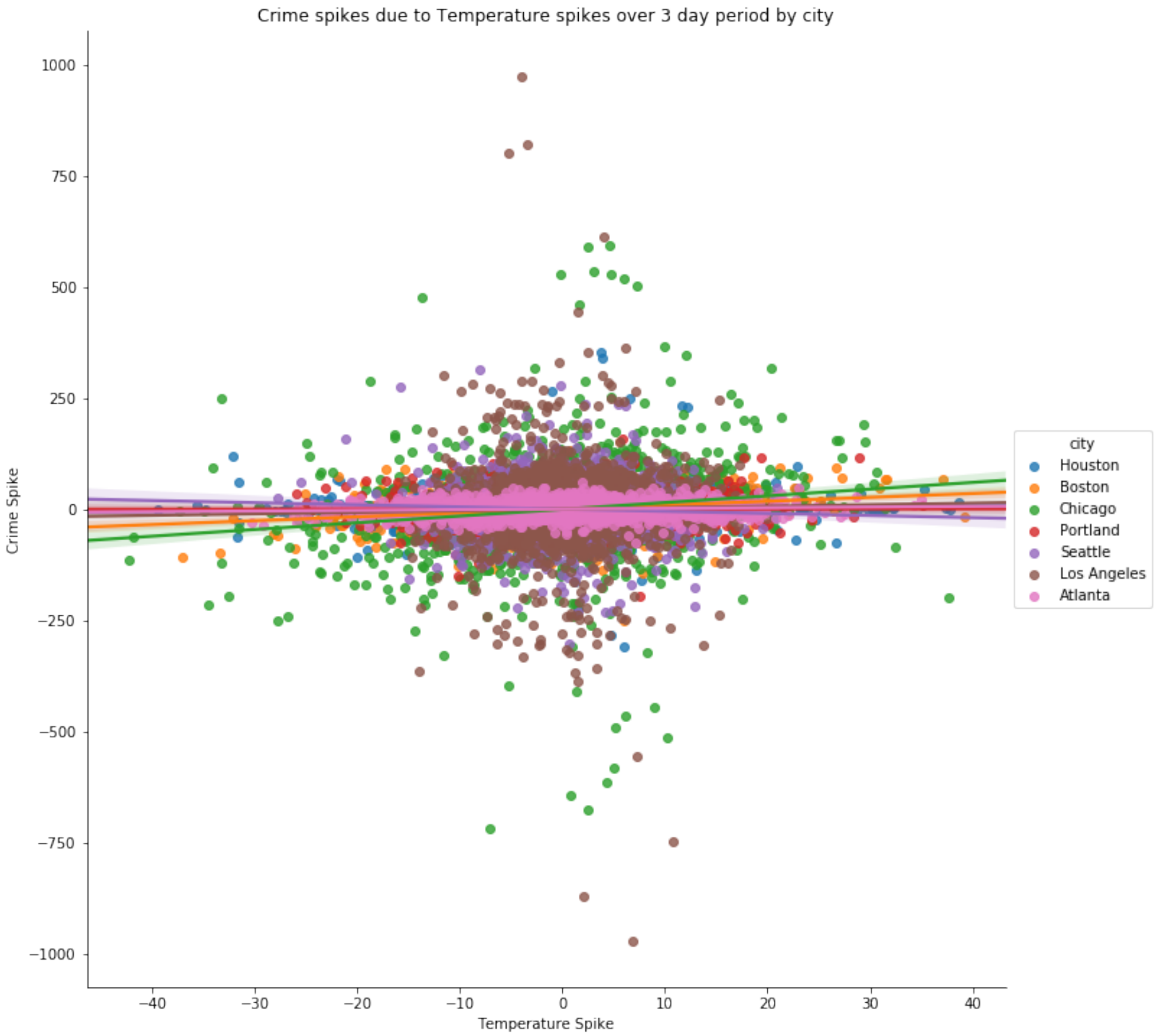
```
g = sns.lmplot(y='Offense_diff', data = shift_merge, x='temp_diff', hu
e = 'Offense_Num',\
               size = 10, aspect = 1, palette = sns.color_palette("Set
1", n_colors=13))
g.set(xlabel = 'Temperature Spike ',ylabel ='Crime Spike',title='Crime
spikes due to \
Temperature spikes over 3 day period by Offense Types')
legend =g._legend
for t,l in zip(legend.texts,('Theft','Burglary','Assault','Sex Crimes'
,'Forgery','Weapons','Drugs','Fraud',\
                             'Arson','Kidnapping','Homicide','Harassmen
t','Driving')):
    t.set_text(l)
```

```
Out[701]: <seaborn.axisgrid.FacetGrid at 0x1b2dbac4e0>
```



Temperature spikes vs. Offense spikes per City:

```
In [702]: f=sns.lmplot(y='Offense_diff', data = shift_merge_cities, x='temp_diff
          ' ,  hue = 'city', size = 10, aspect = 1)
          f.set(xlabel = 'Temperature Spike',ylabel ='Crime Spike',title='Crime
          spikes due to \
          Temperature spikes over 3 day period by city')

Out[702]: <seaborn.axisgrid.FacetGrid at 0x1b5b461ef0>
```



Managerial Insight:

Police departments can use this insight to focus their monitoring efforts in impoverished areas where air conditioning or heating may not be a standard feature in homes, due to which sudden temperature spikes may have a more dramatic effect. This is also an area in which the government can step in to provide water or facilities where people can shelter themselves from the heat or cold in order to prevent the emotional responses that lead to the increase in assaults.

Machine Learning:

In [703]:

```
shift_merge_clean = shift_merge.dropna(how='any', axis=0)
```

In [704]:

```
def posneg(row):
    if (row <= -100) & (row > -200):
        return 1
    elif (row <= 0) & (row > -100):
        return 2
    elif (row > 0) & (row <= 100):
        return 3
    elif (row > 100) & (row <= 200):
        return 4
```

In [705]:

```
def tempseries(row):
    if (row >= -60) & (row < -40):
        return 1
    elif (row >= -40) & (row < -20):
        return 2
    elif (row >= -20) & (row < 0):
        return 3
    elif (row >= 0) & (row < 21):
        return 4
    elif (row >= 21) & (row < 40):
        return 5
    elif (row >= 40) & (row < 60):
        return 6
    elif (row >= 60) & (row < 80):
        return 7
```

In [706]:

```
shift_merge_clean['temp_bin'] = shift_merge_clean.temp_diff.apply(tempseries)
```

In [707]:

```
shift_merge_clean['pos/neg_off'] = shift_merge_clean.Offense_diff.apply(posneg)
```

```
In [708]: shift_merge_clean.head()
```

Out[708]:

	index	Date	Offense_Num	Temp	Offense_count	temp_shift	Offense_shift	t
0	5768	2014-01-21	1	37.956993	769	64.555507	692.0	-2
1	22624	2017-11-09	1	47.972117	103	66.581774	124.0	-1
2	22271	2017-10-01	1	60.675145	110	77.396449	95.0	-1
3	1293	2013-01-15	1	31.997512	785	46.354335	674.0	-1
4	1508	2013-02-02	1	34.255408	690	48.434466	739.0	-1

```
In [709]: import sklearn as sk
          import sklearn.tree as tree
          from IPython.display import Image
          import pydotplus
```

```
In [710]: dt = tree.DecisionTreeClassifier(max_depth=4)
```

```
In [711]: shift_merge_clean = shift_merge_clean.dropna(how='any', axis=0)
```

```
In [712]: X = shift_merge_clean.drop(['temp_diff','pos/neg_off', 'Offense_shift'
          , 'temp_shift', \
                                        'Offense_count', 'Temp', 'Date', 'Offense_
          diff', 'Offense_Num', 'index'],axis=1)
```

```
In [713]: Y = shift_merge_clean['pos/neg_off']
```

Build the tree

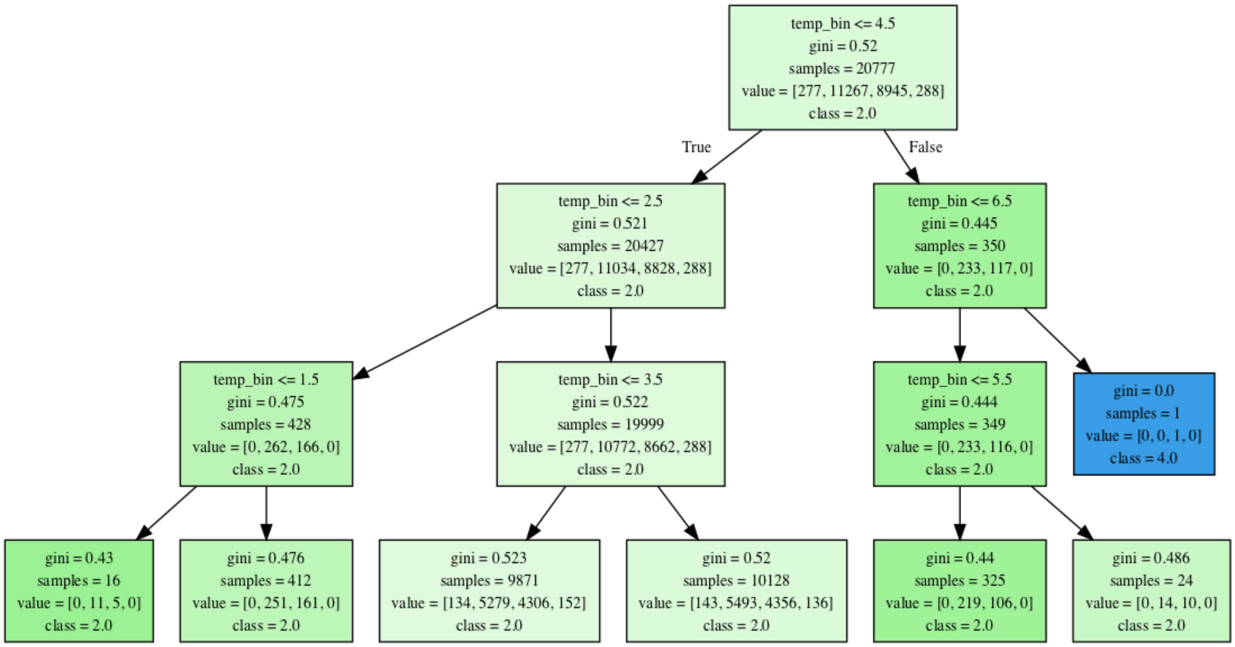
```
In [714]: dt.fit(X,Y)

Out[714]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_dept
          h=4,
                      max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort=False, random_stat
          e=None,
                      splitter='best')
```

Visualize the tree

```
In [715]: len(shift_merge_clean)

Out[715]: 20777
```

```
In [716]: # This code will visualize a decision tree dt,
          # trained with the attributes in X and the class labels in Y
          dt_feature_names = list(X.columns)
          dt_target_names = np.array(Y.unique(),dtype=np.string_)
          result = np.array([s.decode('UTF-8') for s in dt_target_names])
          tree.export_graphviz(dt, out_file='tree.dot',
              feature_names=dt_feature_names, class_names=result,filled=True)
          graph = pydotplus.graph_from_dot_file('tree.dot')
          Image(graph.create_png())
```

Out[716]:



Machine Learning Conclusion:

The decision tree above predicts the amount that crimes will increase or decreases given different spikes in temperature over a three day period. It does not seem to be consistent with the findings in our charts. It can be concluded that temperature spikes alone are not a good predictor of changes in criminal activity.

THANK YOU