Python Data Analysis and Machine Learning

Saman Safari D.

samsafari999@GMAIL.COM

1. Background

1.1 Kaggle Competitions

Kaggle competition is a data science competition where data from different companies and public institutes are provided for analysis. In Kaggle, we choose a particular competition and we download its training datasets. Next step is, building a model using our choice of methods or tools. Finally, the predications are uploaded and Kaggle scores the solution, which are then displayed on the leaderboard. https://www.kaggle.com/competitions (https://www.kaggle.com/competitions)

1.2 San Francisco Crime Classification

"From 1934 to 1963, San Francisco was infamous for housing some of the world's most notorious criminals on the inescapable island of Alcatraz. Today, the city is known more for its tech scene than its criminal past. But, with rising wealth inequality, housing shortages and a proliferation of expensive digital toys riding BART to work, there is no scarcity of crime in the city by the bay."

SF OpenData (https://data.sfgov.org (https://data.sfgov.org)), has provided nearly 12 years of crime reports from across all of San Francisco's neighborhoods. In this datasets, categories of crimes based on occurance time (between 1/1/2003 to 5/13/2015) and location(Geographic coordination system) are given. The data are devived to the training set and test set in a way that weeks 1,3,5,7... belong to test set and weeks 2,4,6,8 belong to training set.

In this competition, we are expected to predict the category of crimes that occurred in the city. I participated in the comepetition and here, I am going to show you my proposed algorithms, codes and results.

2. Introduction

2.1 Envirionment, Python 2.7.

Python is one of the most popular programming languages for data analysis and machine learningdue to it's large number of useful add-on libraries. One might think that the performance of interpreted languages, such as Python, for computation-intensive tasks is inferior to lower-level programming languages, however libraries such as NumPy and SciPy have been developed that build upon lower layer Fortran and C implementations for fast and vectorized operations on multidimensional arrays. Furthermore, for machine learning programming tasks, the scikit-learn library of python is one of the most popular and accessible open source machine learning libraries as of today. For this project, I will be using the following libararies:

Pandas

Providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data. http://pandas.pydata.org (http://pandas.pydata.org)

Sklearn

Features various classification, regression and clustering algorithms, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. http://scikit-learn.org/stable/ (http://scikit-learn.org/stable/)

Numpy

NumPy is the fundamental package for scientific computing with Python. It contains among other things: a powerful N-dimensional array object sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities. NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. http://www.numpy.org (http://www.numpy.org)

Matplotlib

A python plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB®* or Mathematica®†), and web application servers. http://matplotlib.org (http://matplotlib.org)

Seaborn

Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.https://web.stanford.edu/~mwaskom/software/seaborn/# (https://web.stanford.edu/~mwaskom/software/seaborn/#)

2.2 Installation of Python libraries

Download python https://www.python.org/downloads/ (https://www.python.org/downloads/) and install all Pandas, Sklearn, Numpy, and Matplotlib. You may install pip(a tool to install other libraries) and then use the command install For more details about he instructions, refer to https://docs.python.org/2/install/ (https://docs.python.org/2/install/)

Note: Python is available for all three major operating systems—Microsoft Windows, Mac OS X and Linux.

3. Data Analysis

3.1 Collect data

https://www.kaggle.com/c/sf-crime/data (https://www.kaggle.com/c/sf-crime/data) The data are in ".CSV format" and has 2 parts, Train and Test data:

- Train.csv: - models are trained using this dataset.
- Test.csv: - the trained model will use the Test data to predict the category of crimes that occurred.

Data description:

- Dates - timestamp of the crime incident
- Category - category of the crime incident (only in train.csv). This is the target variable you are going to predict.
- Descript - detailed description of the crime incident (only in train.csv)
- DayOfWeek - the day of the week
- PdDistrict - name of the Police Department District
- Resolution - how the crime incident was resolved (only in train.csv)
- Address - the approximate street address of the crime incident
- X - Longitude
- Y - Latitude

3.2 Importing Data

In [1]:

```
import pandas as pd

#Load Data, 'Dates' is in String format, parse_dates to convert it
into 'date time format'.
train = pd.read_csv("/Users/Saman/Desktop/BigD/Kaggle/San_Francisco
_Crime_Classification/train.csv",parse_dates = ['Dates'])
test = pd.read_csv("/Users/Saman/Desktop/BigD/Kaggle/San_Francisco_
Crime_Classification/test.csv", parse_dates = ['Dates'])
print train.head()
```

```
                Dates        Category                      Descrip
t  \
0 2015-05-13 23:53:00        WARRANTS                WARRANT ARRES
T
1 2015-05-13 23:53:00  OTHER OFFENSES      TRAFFIC VIOLATION ARRES
T
2 2015-05-13 23:33:00  OTHER OFFENSES      TRAFFIC VIOLATION ARRES
T
3 2015-05-13 23:30:00   LARCENY/THEFT  GRAND THEFT FROM LOCKED AUT
O
4 2015-05-13 23:30:00   LARCENY/THEFT  GRAND THEFT FROM LOCKED AUT
O

   DayOfWeek PdDistrict      Resolution                    Address
\
0  Wednesday   NORTHERN  ARREST, BOOKED         OAK ST / LAGUNA ST
1  Wednesday   NORTHERN  ARREST, BOOKED         OAK ST / LAGUNA ST
2  Wednesday   NORTHERN  ARREST, BOOKED  VANNESS AV / GREENWICH ST
3  Wednesday   NORTHERN            NONE   1500 Block of LOMBARD ST
4  Wednesday       PARK            NONE  100 Block of BRODERICK ST

            X          Y
0 -122.425892  37.774599
1 -122.425892  37.774599
2 -122.424363  37.800414
3 -122.426995  37.800873
4 -122.438738  37.771541
```

3.3 Handling missing data

We search for missing data in our Train and Test datasets. The { isnull().sum(axis=0)} provides us with the total number of missing data across all columns. If axis=1, missing data across the rows will be showed. It can be seen that our datasets are clean and no 'NaN' is observed.

```
In [2]: print 'train missing data:\n', train.isnull().sum(axis=0)
        print 'test missing data:\n', test.isnull().sum(axis=0)
```

```
train missing data:
Dates          0
Category       0
Descript       0
DayOfWeek      0
PdDistrict     0
Resolution     0
Address        0
X              0
Y              0
dtype: int64
test missing data:
Id             0
Dates          0
DayOfWeek      0
PdDistrict     0
Address        0
X              0
Y              0
dtype: int64
```

3.4 Mining and & Refining the data

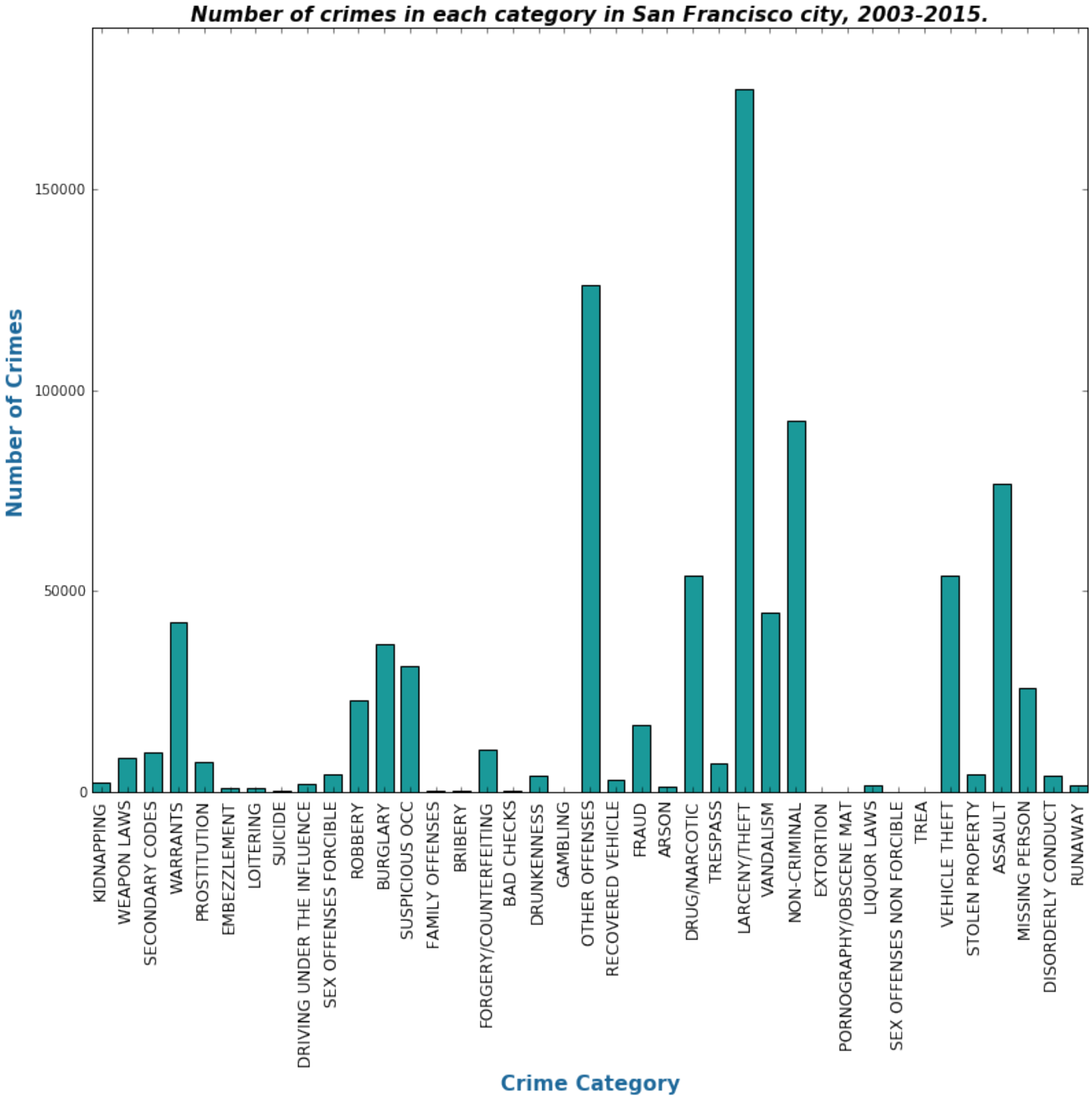
In this step, we will have an overall view of our data to identify trends. Descriptive analysis and visulizations are effective ways to find out outliers and to get an overall picture of our datasets.

3.4.1 Number of crimes in each category

Figure 1 depicts the number of crimes in each category in San Francisco city between 2003-2015. It can be seen that crime categories such as LARCENY/THEFT, OTHER OFFENSES and NON-CRIMINAL are the three most common crimes across the city. TREA is the least common crime with 6 occurrence frequency.

In [3]:

```
%matplotlib inline


import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-20,370,39)

y = np.array([2341, 8555, 9985, 42214, 7484, 1166, 1225, 508, 2268,
4388, 23000, 36755, 31414, 491, 289,
            10609, 406, 4280, 146, 126182, 3138, 16679, 1513, 53971
, 7326, 174900, 44725, 92304, 256, 22,
            1903, 148, 6, 53781, 4540, 76876, 25989, 4320, 1946])



my_xticks = ['KIDNAPPING','WEAPON LAWS','SECONDARY CODES','WARRANTS
', 'PROSTITUTION','EMBEZZLEMENT', 'LOITERING',
            'SUICIDE', 'DRIVING UNDER THE INFLUENCE','SEX OFFENSES
FORCIBLE','ROBBERY', 'BURGLARY',
            'SUSPICIOUS OCC','FAMILY OFFENSES', 'BRIBERY', 'FORGERY
/COUNTERFEITING', 'BAD CHECKS', 'DRUNKENNESS',
            'GAMBLING', 'OTHER OFFENSES','RECOVERED VEHICLE','FRAUD
','ARSON','DRUG/NARCOTIC', 'TRESPASS',
            'LARCENY/THEFT','VANDALISM', 'NON-CRIMINAL', 'EXTORTION
', 'PORNOGRAPHY/OBSCENE MAT','LIQUOR LAWS',
            'SEX OFFENSES NON FORCIBLE','TREA', 'VEHICLE THEFT','ST
OLEN PROPERTY','ASSAULT','MISSING PERSON',
            'DISORDERLY CONDUCT','RUNAWAY']




plt.figure(figsize=(13,10))
plt.xticks(x, my_xticks, rotation=90, fontsize=12)
plt.ylim(0,190000)
#plt.xlim(-5,350)
plt.bar(x, y,width=7, color = (.1, 0.6, 0.6), align='center')
plt.ylabel('Number of Crimes', fontsize=15, color = (.1, 0.4, 0.6),
fontweight='bold')
plt.xlabel('Crime Category', fontsize=15, color = (.1, 0.4, 0.6), f
ontweight='bold')
plt.title('Number of crimes in each category in San Francisco city,
2003-2015.', style='italic', fontsize=15,
          fontweight='bold')
plt.savefig("plot_1.png")
plt.show()
```

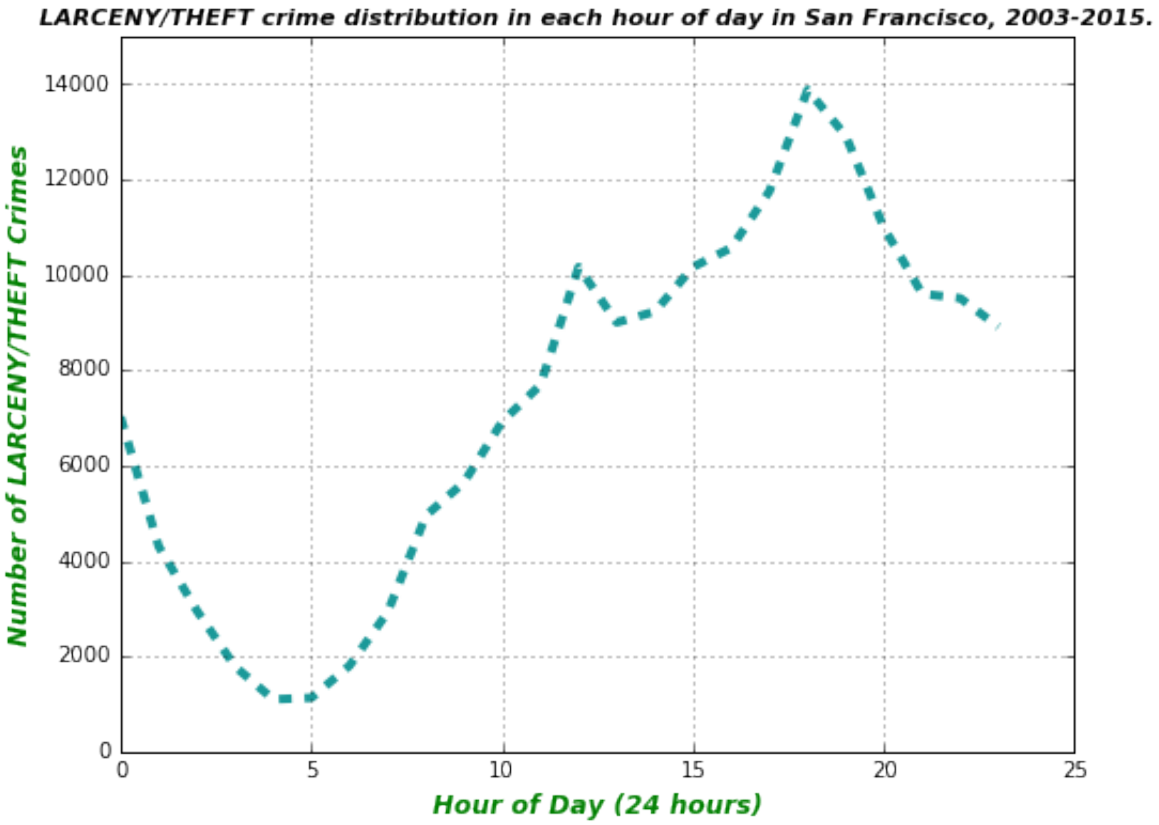


3.4.2 At what time a day 'LARCENY/THEFT' happens?

As Figure 1 depicted, LARCENY/THEFT is the most common crime in the city. Figure 2 demonstrates the number of LARCENY/THEFT crime vs. hour of a day. As shown, maximum number of LARCENY/THEFT occurs around 19 (7:00 PM) with occurance frequency of 12912. On the other hand, minimum number of crimes occured around 4:00 AM (1098 crime). It can be concluded that the possibility of LARCENY/THEFT occurance around 7:00 PM and 4:00 AM are maximum and minimum, repectively.

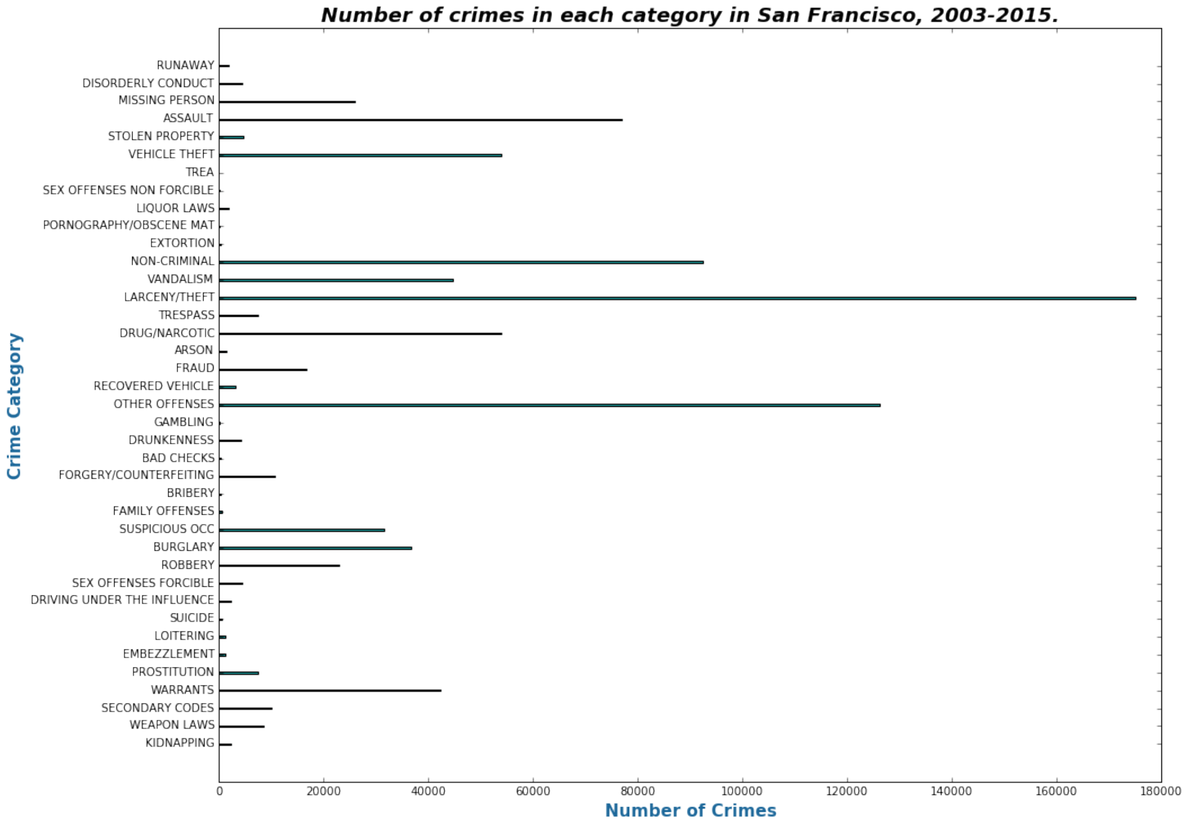
In [6]:

```
'''
tt = len(train.Category)
tim = train.Dates.dt.hour
co = []
for i in range(tt):
    if train.Category[i]=='LARCENY/THEFT':
        co.append(tim[i])
    else:
         x = 'no'
print Counter(co)
'''



x = np.array([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
21,22,23])
y = np.array([7019, 4304, 2957, 1786, 1098, 1130, 1806, 2930, 4952,
5650, 6924, 7688, 10160, 8999, 9229, 10164, 10564,
    11753, 13875, 12912, 10984, 9600, 9507, 8909])

plt.figure(figsize=(8,6))
plt.plot(x,y, color = (.1, 0.6, 0.6), linestyle = '--', linewidth =
4)
plt.grid(True)
#plt.color('red')

plt.ylim(0,15000)
plt.ylabel('Number of LARCENY/THEFT Crimes', fontsize=12, color='gr
een', fontweight='bold', style='italic')
plt.xlabel('Hour of Day (24 hours)', fontsize=12, color='green', fo
ntweight='bold', style='italic')
plt.title('LARCENY/THEFT crime distribution in each hour of day in
San Francisco, 2003-2015.', style='italic',
            fontsize=11, fontweight='bold')



plt.savefig("plot_2.png")

plt.show()
```


LARCENY/THEFT crime distribution in each hour of day in San Francisco, 2003-2015.
Number of LARCENY/THEFT Crimes
Hour of Day (24 hours)
0
2000
4000
6000
8000
10000
12000
14000
0
5
10
15
20
25

In [8]:

```
y = np.linspace(20,380,39)


x = np.array([2341, 8555, 9985, 42214, 7484, 1166, 1225, 508, 2268,
4388, 23000, 36755, 31414, 491, 289,
            10609, 406, 4280, 146, 126182, 3138, 16679, 1513, 53971
, 7326, 174900, 44725, 92304, 256, 22,
            1903, 148, 6, 53781, 4540, 76876, 25989, 4320, 1946])


my_yticks = ['KIDNAPPING','WEAPON LAWS','SECONDARY CODES','WARRANTS
', 'PROSTITUTION','EMBEZZLEMENT', 'LOITERING',
            'SUICIDE', 'DRIVING UNDER THE INFLUENCE','SEX OFFENSES
FORCIBLE','ROBBERY', 'BURGLARY',
            'SUSPICIOUS OCC','FAMILY OFFENSES', 'BRIBERY', 'FORGERY
/COUNTERFEITING', 'BAD CHECKS', 'DRUNKENNESS',
            'GAMBLING', 'OTHER OFFENSES','RECOVERED VEHICLE','FRAUD
','ARSON','DRUG/NARCOTIC', 'TRESPASS',
            'LARCENY/THEFT','VANDALISM', 'NON-CRIMINAL', 'EXTORTION
', 'PORNOGRAPHY/OBSCENE MAT','LIQUOR LAWS',
            'SEX OFFENSES NON FORCIBLE','TREA', 'VEHICLE THEFT','ST
OLEN PROPERTY','ASSAULT','MISSING PERSON',
            'DISORDERLY CONDUCT','RUNAWAY']

plt.figure(figsize=(15,12))
plt.barh(y,x, color = (.1, 0.6, 0.6), align='center')
plt.yticks(y, my_yticks)
plt.ylabel('Crime Category', fontsize=15, color = (.1, 0.4, 0.6), f
ontweight='bold')
plt.xlabel('Number of Crimes', fontsize=15, color = (.1, 0.4, 0.6),
fontweight='bold')
plt.title('Number of crimes in each category in San Francisco, 2003
-2015.', style='italic', fontsize=18, fontweight='bold')

plt.savefig("plot_3.png")
plt.show()
```

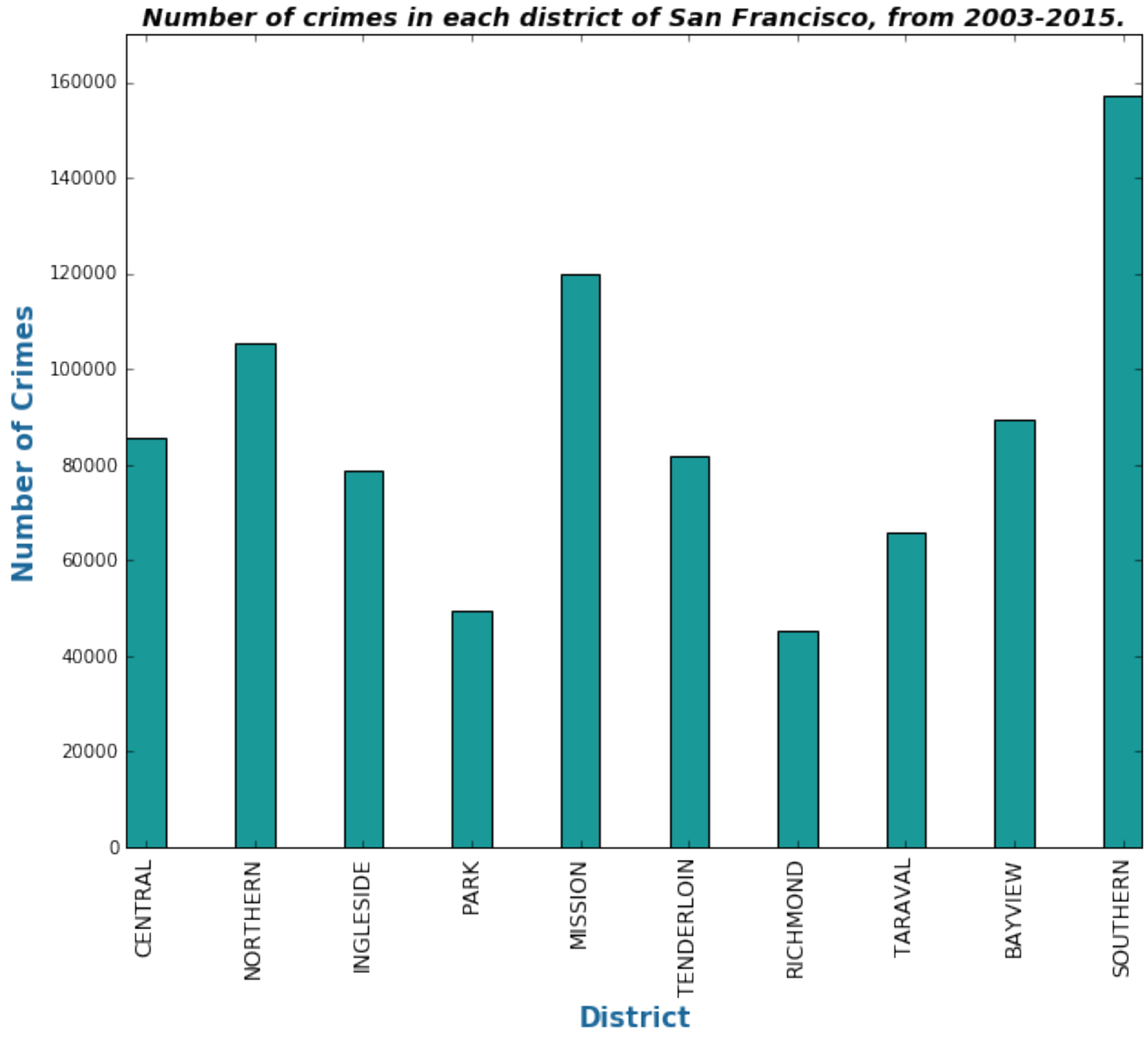


3.4.3 Crime distribution across different district

According to our datasets, San Francisco is devided into 10 districts including 'CENTRAL','NORTHERN','INGLESIDE', 'PARK','MISSION','TENDERLOIN', 'RICHMOND','TARAVAL', 'BAYVIEW', and 'SOUTHERN'. For different socioeconomic reasons, crime distribution across the districst varies. Figure 4 shows that maximum number of crimes occurred in SOUTHERN district followed by MISSION district and the least number of crimes occurred in RICHMOND district.

In [11]:

```
x = np.linspace(0,100,10)
y = np.array([85460, 105296, 78845, 49313, 119908, 81809, 45209, 65
596, 89431, 157182])



my_xticks = ['CENTRAL','NORTHERN','INGLESIDE', 'PARK','MISSION','TE
NDERLOIN', 'RICHMOND','TARAVAL', 'BAYVIEW',
             'SOUTHERN']




plt.figure(figsize=(10,8))
plt.xticks(x, my_xticks, rotation=90, fontsize=12)
plt.ylim(0,170000)
#plt.xlim(-5,350)
plt.bar(x, y,width=4, color = (.1, 0.6, 0.6), align='center')
plt.ylabel('Number of Crimes', fontsize=15, color = (.1, 0.4, 0.6),
fontweight='bold')
plt.xlabel('District', fontsize=15, color = (.1, 0.4, 0.6), fontwei
ght='bold')
plt.title('Number of crimes in each district of San Francisco, from
2003-2015.', style='italic', fontsize=14,
          fontweight='bold')

plt.savefig("plot_4.png")
plt.show()
```

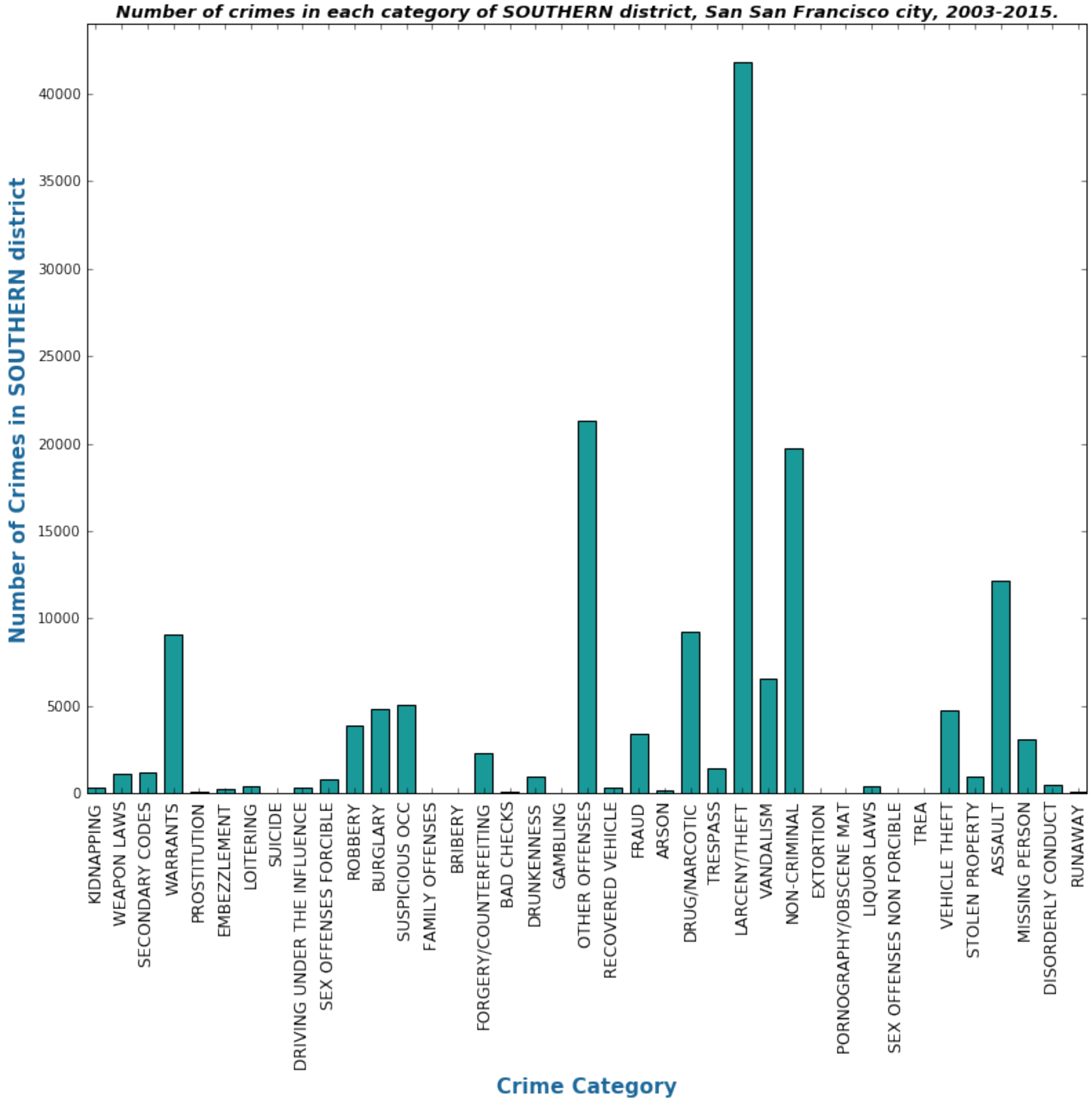


SOUTHERN district attracts our attention because of its extreemly high number of crimes compared to the other districts. It is noteworthy to study SOUTHERN district in more details. For instance, what category of crimes occurs most commonly in SOUTHERN. Figure 5 shows the number of crimes in each category that occurred in SOUTHERN district. It is clear that the most common crime is LARCENY/THEFT. Furthermore, Figure 5 depicts that the trend in crime category in SOUTHERN district is very simmilar to overall crime's trend across the San Francisco city {Figure 1}.

In [15]:

```
'''
tt = len(train.PdDistrict)
pp = train.PdDistrict
cat = train.Category
Counter(train.PdDistrict)
#tim = train.Dates.dt.hour
co = []
for i in range(tt):
    if pp[i]=='SOUTHERN':
        co.append(cat[i])
    else:
         x = 'no'
print Counter(co)
```

```
'''

x = np.linspace(-20,370,39)

y = np.array([340, 1128, 1205, 9102, 135, 275, 429, 59, 306, 819, 3875, 4841, 5065, 42, 37,
            2345, 74, 959, 17, 21308, 326, 3441, 185, 9228, 1456, 41845, 6550, 19745, 38, 4,
            385, 17, 0, 4725, 1007, 12183, 3064, 511, 108])


my_xticks = ['KIDNAPPING','WEAPON LAWS','SECONDARY CODES','WARRANTS', 'PROSTITUTION','EMBEZZLEMENT', 'LOITERING',
            'SUICIDE', 'DRIVING UNDER THE INFLUENCE','SEX OFFENSES FORCIBLE','ROBBERY', 'BURGLARY',
            'SUSPICIOUS OCC','FAMILY OFFENSES', 'BRIBERY', 'FORGERY/COUNTERFEITING', 'BAD CHECKS', 'DRUNKENNESS',
            'GAMBLING', 'OTHER OFFENSES','RECOVERED VEHICLE','FRAUD','ARSON','DRUG/NARCOTIC', 'TRESPASS',
            'LARCENY/THEFT','VANDALISM', 'NON-CRIMINAL', 'EXTORTION', 'PORNOGRAPHY/OBSCENE MAT','LIQUOR LAWS',
            'SEX OFFENSES NON FORCIBLE','TREA', 'VEHICLE THEFT','STOLEN PROPERTY','ASSAULT','MISSING PERSON',
            'DISORDERLY CONDUCT','RUNAWAY']



plt.figure(figsize=(13,10))
plt.xticks(x, my_xticks, rotation=90, fontsize=12)
plt.ylim(0,44000)
#plt.xlim(-5,350)
plt.bar(x, y,width=7, color = (.1, 0.6, 0.6), align='center')
plt.ylabel('Number of Crimes in SOUTHERN district', fontsize=15, color = (.1, 0.4, 0.6), fontweight='bold')
plt.xlabel('Crime Category', fontsize=15, color = (.1, 0.4, 0.6), fontweight='bold')
plt.title('Number of crimes in each category of SOUTHERN district, San San Francisco city, 2003-2015.',
          style='italic', fontsize=13,
          fontweight='bold')

plt.savefig("plot_5.png")
plt.show()
```

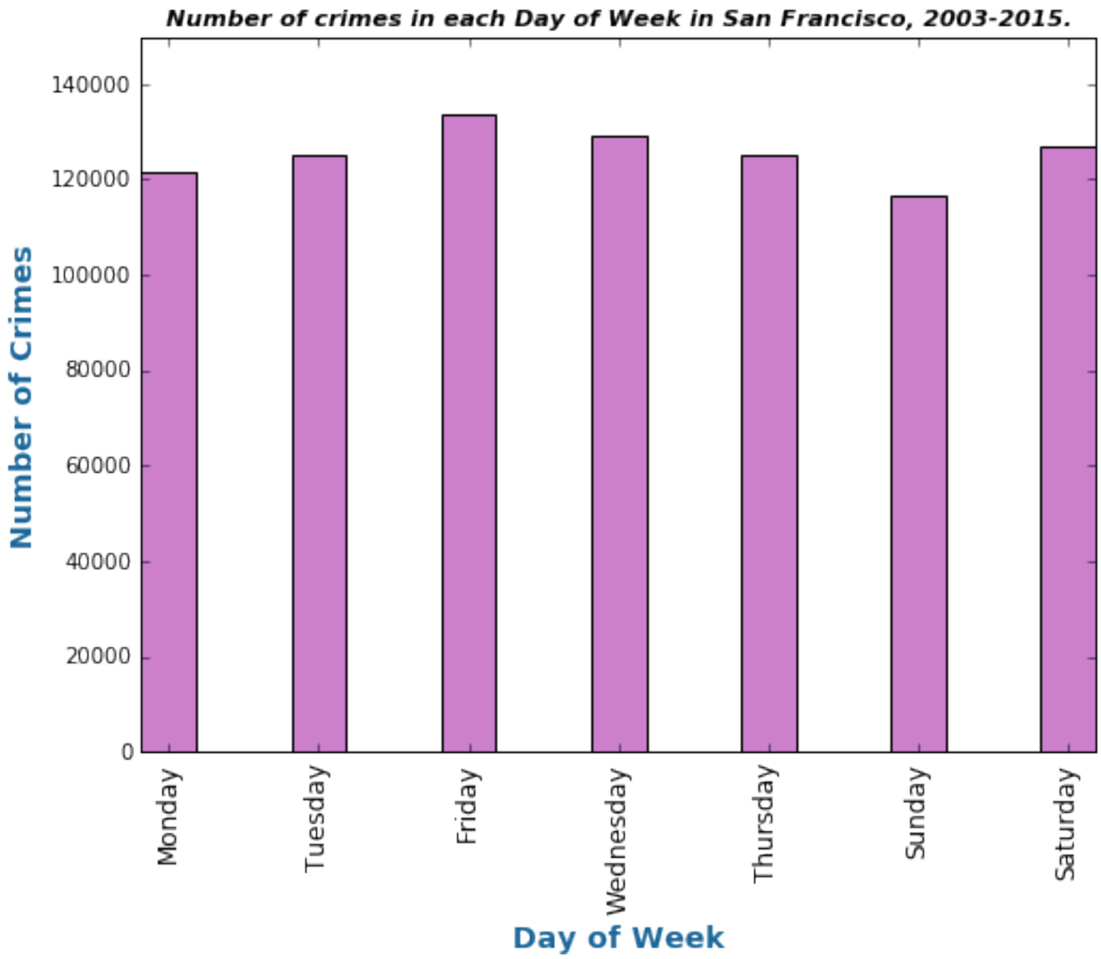


3.4.4 Crime distribution across Day of Week

Figure 6 shows crime distribution across Day of Week including Monday','Tuesday','Friday','Wednesday','Thursday','Sunday', and 'Saturday'. It can be seen that most crimes occurs on Fridays whereas Sundays have the minimum number of crimes.

In [4]:

```
x = np.linspace(0,49,7)
y = np.array([121584, 124965, 133734, 129211, 125038, 116707, 12681
0])
my_xticks = ['Monday','Tuesday','Friday','Wednesday','Thursday','Su
nday','Saturday']

plt.figure(figsize=(8,6))
plt.xticks(x, my_xticks, rotation=90, fontsize=12)
plt.ylim(0,150000)
#plt.xlim(-5,350)
plt.bar(x, y,width=3, color = (.8, 0.5, 0.8), align='center')
plt.ylabel('Number of Crimes', fontsize=14, color = (.1, 0.4, 0.6),
fontweight='bold')
plt.xlabel('Day of Week', fontsize=14, color = (.1, 0.4, 0.6), font
weight='bold')
plt.title('Number of crimes in each Day of Week in San Francisco, 2
003-2015.', style='italic',
              fontsize=11, fontweight='bold')

plt.savefig("plot_6.png")
plt.show()
```

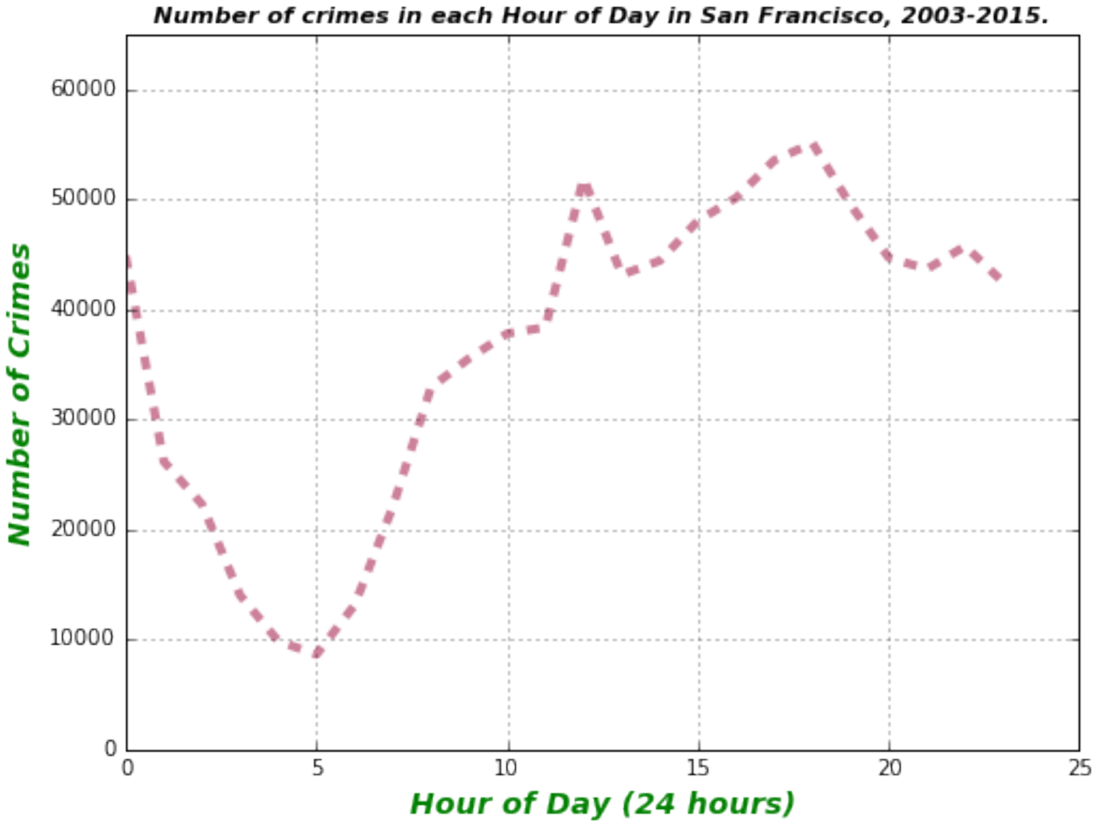


3.4.5 Crime distribution in each Hour of Day

Figure 7 shows the total number of crimes in each hour of day in San Francisco between 2003-2015.In figure 7, horizental axis is based on 24 hours(0-23). It is clear that maximum number of crimes occurred at 18:00(6:00 PM) with a totall number of 55104 and the minimum number of crimes occurred at 5:00 AM with a totall number of 8637. It should be noted that this rsults is different from Figure 2 in which the LARCENY/THEFT crime category was studied. In figure 2, the LARCENY/THEFT occurance around 7:00 PM and 4:00 AM are maximum and minimum, repectively.

In [5]:

```
crime_hour = Counter(train.Dates.dt.hour)
x = list(crime_hour)
y = []
for i in x:
    crime_co = crime_hour[i]
    y.append(crime_co)

plt.figure(figsize=(8,6))
plt.plot(x,y, color = (.8, 0.5, 0.6), linestyle = '--', linewidth =
4)
plt.grid(True)

plt.ylim(0,65000)
plt.ylabel('Number of Crimes', fontsize=14, color='green', fontweig
ht='bold', style='italic')
plt.xlabel('Hour of Day (24 hours)', fontsize=14, color='green', fo
ntweight='bold', style='italic')
plt.title('Number of crimes in each Hour of Day in San Francisco, 2
003-2015.', style='italic',
            fontsize=11, fontweight='bold')




plt.savefig("plot_7.png")

plt.show()
```



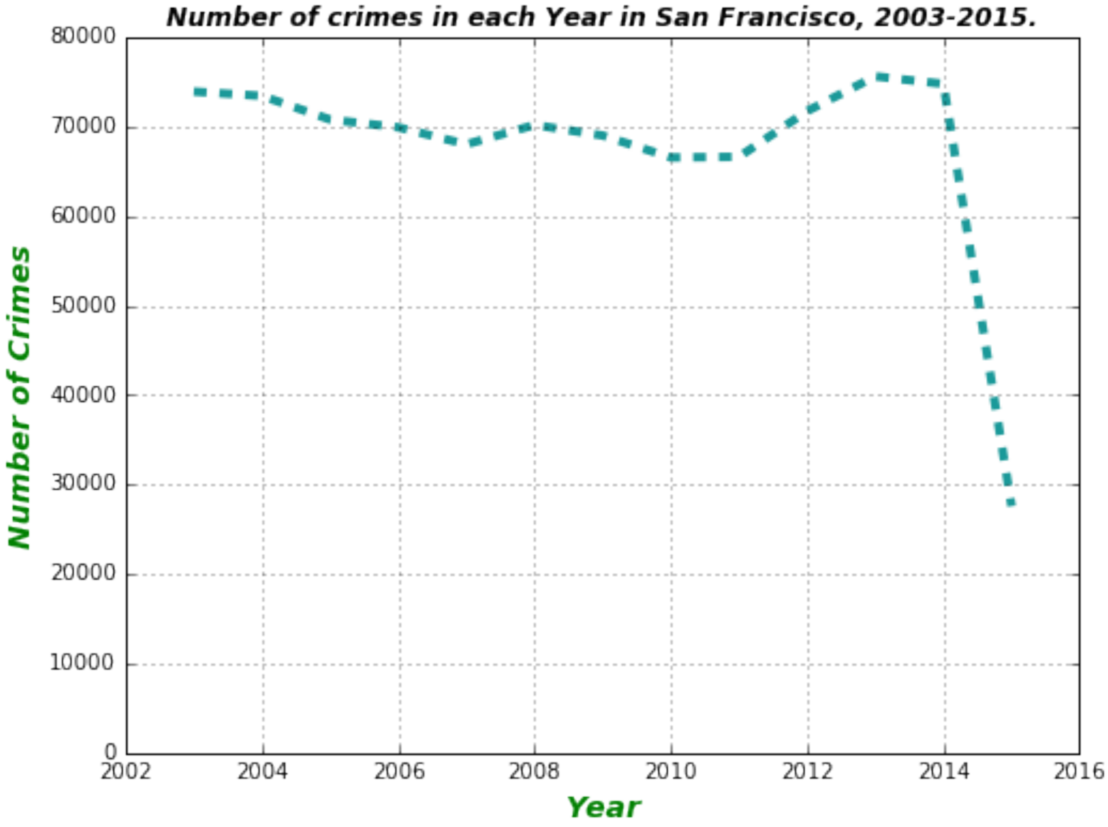
3.4.6 Number of crimes in each Year, from 2003-2015

The number of crimes in each year from 2003-2015 in San Francisco is shown in Figure 8. Between 2003-2007 the number of crimes decreased from 75606 in 2003 to 68015 in 2007. However, in 2008 the number of crimes started to rise to 70174 and followed by a decrease to 66619 in 2011. After 2011, the number of crimes started to increase again and reached to a maximum level of 75606, the highest number from 2003-2015. The sharp drop in 2015 is related to incomplete dataset for 2015.

In [6]:

```
crime_year = Counter(train.Dates.dt.year)
x = list(crime_year)
print crime_year
y = []
for i in x:
    crime_co = crime_year[i]
    y.append(crime_co)

plt.figure(figsize=(8,6))
plt.plot(x,y, color = (.1, 0.6, 0.6), linestyle = '--', linewidth =
4)
plt.grid(True)

plt.ylim(0,80000)
plt.ylabel('Number of Crimes', fontsize=14, color='green', fontweig
ht='bold', style='italic')
plt.xlabel('Year', fontsize=14, color='green', fontweight='bold', s
tyle='italic')
plt.title('Number of crimes in each Year in San Francisco, 2003-201
5.', style='italic',
            fontsize=12, fontweight='bold')




plt.savefig("plot_8.png")

plt.show()
```

```
Counter({2013: 75606, 2014: 74766, 2003: 73902, 2004: 73422, 2012:
71731, 2005: 70779, 2008: 70174, 2006: 69909, 2009: 69000, 2007: 6
8015, 2011: 66619, 2010: 66542, 2015: 27584})
```

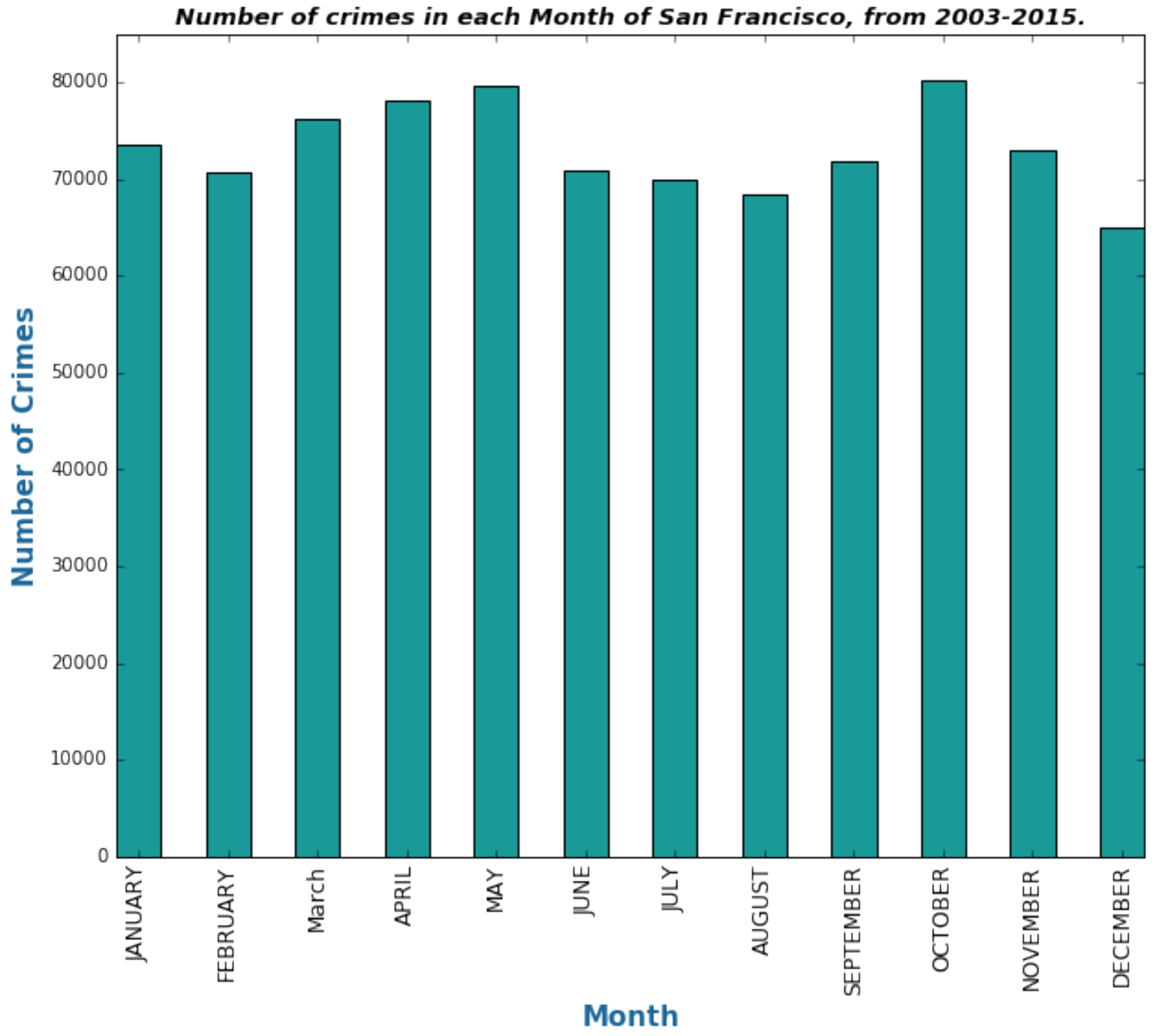


3.4.7 Number of crimes in each Month, from 2003-2015

Figure 9 shows the number of crimes in each month over the course of years between 2003-2015. It can be seen that the maximum number of crimes, 80274, occurred in OCTOBER and the minimum number of crimes, 65006, occurred in DECEMBER.

In [7]:

```
crime_month = Counter(train.Dates.dt.month)
x = list(crime_month)

y = []
for i in x:
    crime_co = crime_month[i]
    y.append(crime_co)

my_xticks = ['JANUARY','FEBRUARY','March', 'APRIL','MAY','JUNE', 'J
ULY','AUGUST', 'SEPTEMBER', 'OCTOBER', 'NOVEMBER',
             'DECEMBER']




plt.figure(figsize=(10,8))
plt.xticks(x, my_xticks, rotation=90, fontsize=12)
plt.ylim(0,85000)
plt.bar(x, y,width=.5, color = (.1, 0.6, 0.6), align='center')
plt.ylabel('Number of Crimes', fontsize=15, color = (.1, 0.4, 0.6),
fontweight='bold')
plt.xlabel('Month', fontsize=15, color = (.1, 0.4, 0.6), fontweight
='bold')
plt.title('Number of crimes in each Month of San Francisco, from 20
03-2015.', style='italic', fontsize=13,
          fontweight='bold')


plt.savefig("plot_9.png")
plt.show()
```



4. Machine Learning - Crime Prediction

In this section, given parameters including Day of Week, District and Category of crimes, we predict the category of crimes that occur in the city. Algorithms such as Logistic Regression, Support Vector Machines(SVM), K-Nearest Neighbors(KNN), Linear discriminant analysis(LDA), and GradientBoostingClassifier. We compare results acheived from each of the classifier.