奥克兰犯罪统计数据集

1. 数据集加载

由于2012与2014年数据集中Location一栏与其他年份格式不同，故对这两个文件进行单独处理。

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
%matplotlib inline

path = "./dataset/oakland_crime_statistics/records-for-{}.csv"
data = pd.concat([pd.read_csv(path.format(year)) for year in [2011,2013,2015,2016]])
data2 = pd.concat([pd.read_csv(path.format(year)) for year in [2012,2014]])
data2['Location 1'] = data2['Location 1'].astype(str)
data2['Location'] = ''
for index,row in data2.iterrows():
    l = row['Location 1'].split("address")
    if len(l)>2:
        l = l[2].split("city")[0].replace('"','').replace(":","").replace(",","")
    else:
        l = None
    row['Location'] = l
```

```python
data2 = data2.drop(columns=["Location 1"])
print('处理完成')
data = pd.concat([data,data2],ignore_index=True)
data.info()
#data.head(5)
#data2.info()
```

```
处理完成
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1046388 entries, 0 to 1046387
Data columns (total 11 columns):
 #   Column                     Non-Null Count    Dtype
---  ------                     --------------    -----
 0   Agency                     1046384 non-null  object
 1   Create Time                1046384 non-null  object
 2   Location                   671477 non-null   object
 3   Area Id                    864023 non-null   object
 4   Beat                       1040583 non-null  object
 5   Priority                   1046384 non-null  float64
 6   Incident Type Id           1046384 non-null  object
 7   Incident Type Description  1045996 non-null  object
 8   Event Number               1046384 non-null  object
 9   Closed Time                1046359 non-null  object
 10  Zip Codes                  352 non-null      float64
dtypes: float64(2), object(9)
```

```
memory usage: 87.8+ MB
```

2. 数据集可视化可摘要

2.1 数据摘要和可视化

- 共13个属性,其中:
- 标称属性:
    1. Agency 办事处
    2. Location 事发区域
    3. Area Id 区域ID
    4. Beat 击败
    5. Priority 优先级
    6. Incident Type Id 事件类型ID
    7. Incident Type Description 事件类型描述
    8. Event Number 事件代码
    9. Zip Codes 邮政编码
- 数值属性:

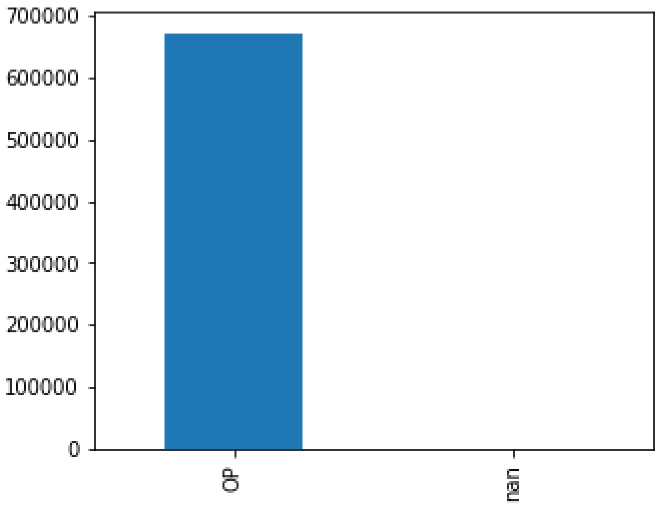
    10. Create Time 发生时间
    11. Closed Time 结束时间

(1) Agency属性

```python
print(data['Agency'].value_counts(dropna = False).head(10))
data['Agency'].value_counts(dropna = False).plot(kind="bar",figsize=(5,4))
```

```
OP     671474
NaN         3
Name: Agency, dtype: int64

<AxesSubplot:>
```

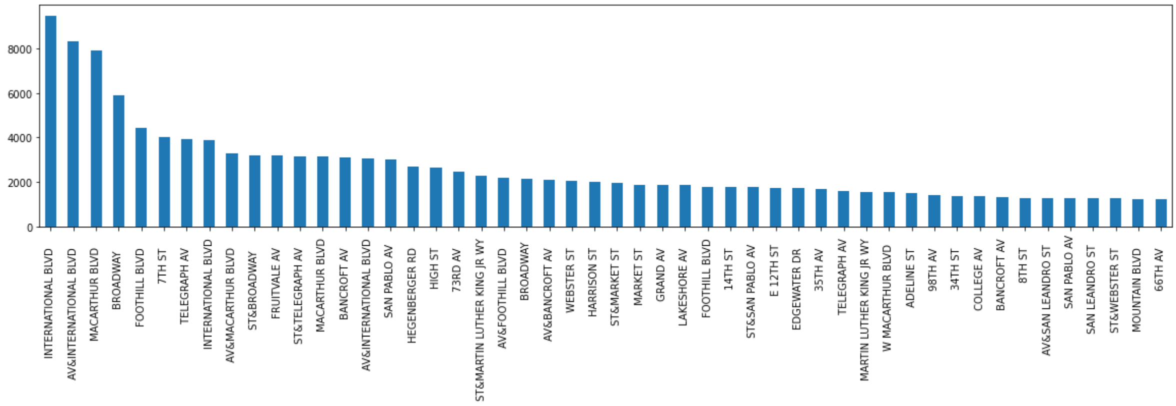


(2) Location属性

```
print(data['Location'].value_counts(dropna = False).head(10))
data['Location'].value_counts(dropna = False)[:50].plot(kind="bar",figsize=
(20,4))
```

```
 INTERNATIONAL BLVD          9498
 AV&INTERNATIONAL BLVD       8340
 MACARTHUR BLVD              7920
 BROADWAY                    5915
 FOOTHILL BLVD               4455
 7TH ST                      4038
 TELEGRAPH AV                3940
 INTERNATIONAL BLVD          3866
 AV&MACARTHUR BLVD           3305
 ST&BROADWAY                 3215
Name: Location, dtype: int64

<AxesSubplot:>
```

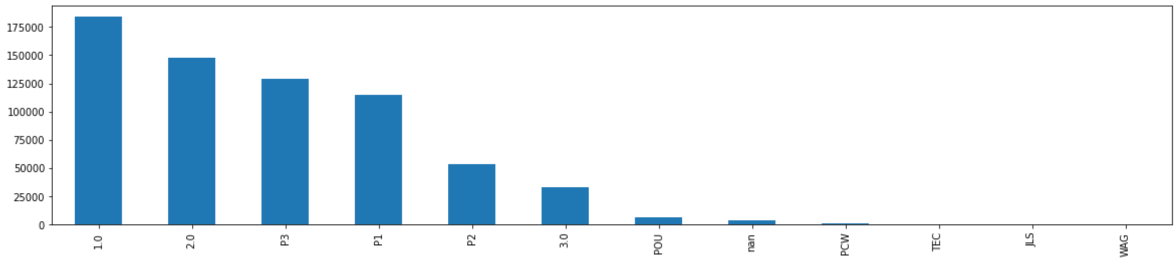


(3) Area Id属性

```
print(data['Area Id'].value_counts(dropna = False).head(10))
data['Area Id'].value_counts(dropna = False).plot(kind="bar",figsize=(20,4))
```

```
1.0    184368
2.0    147839
P3     129054
P1     114560
P2      53033
3.0     32699
POU      5960
NaN      3163
PCW       789
TEC        10
Name: Area Id, dtype: int64

<AxesSubplot:>
```

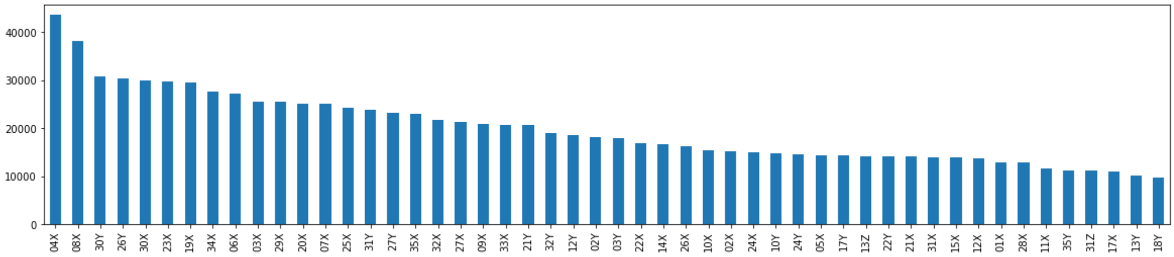


(4) Beat属性

```
print(data['Beat'].value_counts(dropna = False).head(10))
data['Beat'].value_counts(dropna = False)[:50].plot(kind="bar",figsize=
(20,4))
```

```
04X    43626
08X    38097
30Y    30880
26Y    30377
30X    29881
23X    29684
19X    29633
34X    27591
06X    27148
03X    25587
Name: Beat, dtype: int64

<AxesSubplot:>
```

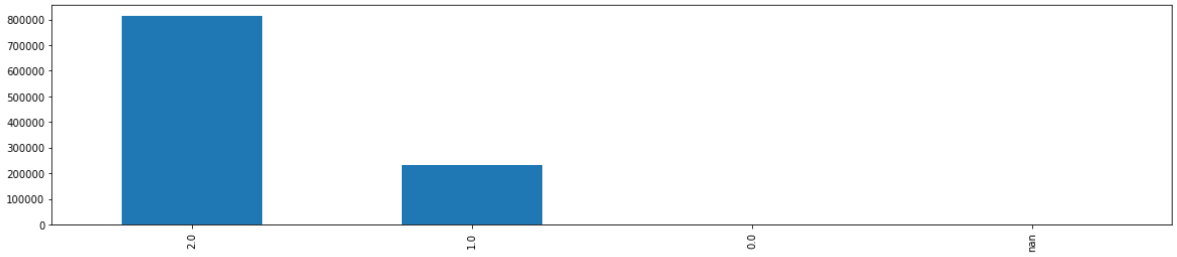


(5) Priority属性

```
print(data['Priority'].value_counts(dropna = False).head(10))
data['Priority'].value_counts(dropna = False).plot(kind="bar",figsize=
(20,4))
```

```
2.0    814818
1.0    231542
0.0        24
NaN         4
Name: Priority, dtype: int64

<AxesSubplot:>
```

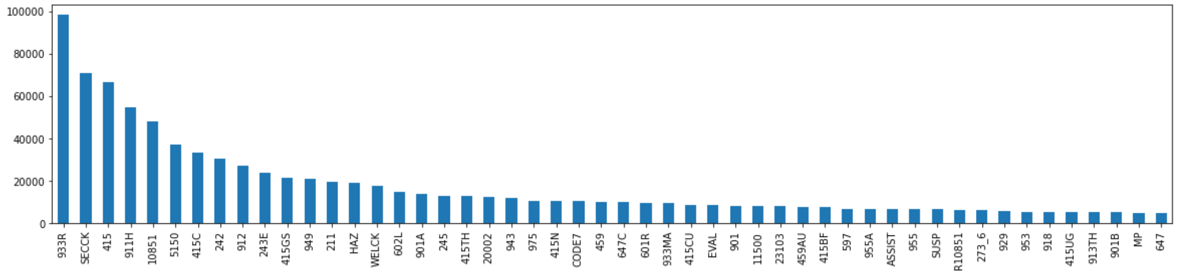


(6) Incident Type Id属性

```python
print(data['Incident Type Id'].value_counts(dropna = False).head(10))
data['Incident Type Id'].value_counts(dropna = False)
[:50].plot(kind="bar",figsize=(20,4))
```

```
933R     98497
SECCK    70965
415      66720
911H     54935
10851    47958
5150     37218
415C     33470
242      30636
912      26984
243E     23964
Name: Incident Type Id, dtype: int64

<AxesSubplot:>
```

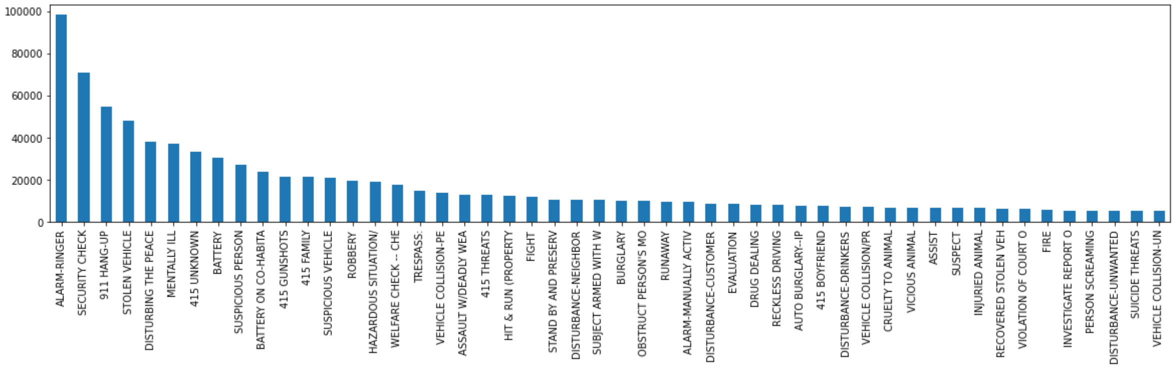


(7) Incident Type Description属性

```python
print(data['Incident Type Description'].value_counts(dropna =
False).head(10))
data['Incident Type Description'].value_counts(dropna = False)
[:50].plot(kind="bar",figsize=(20,4))
```

```
ALARM-RINGER             98497
SECURITY CHECK           70965
911 HANG-UP              54935
STOLEN VEHICLE           47958
DISTURBING THE PEACE     38257
MENTALLY ILL             37218
415 UNKNOWN              33470
BATTERY                  30636
SUSPICIOUS PERSON        26984
BATTERY ON CO-HABITA     23964
Name: Incident Type Description, dtype: int64

<AxesSubplot:>
```

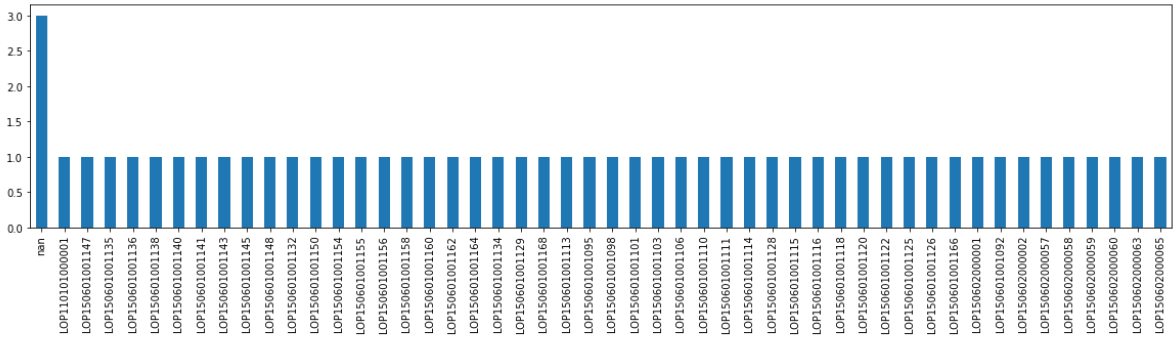


(8) Event Number属性

```
print(data['Event Number'].value_counts(dropna = False).head(10))
data['Event Number'].value_counts(dropna = False)[:50].plot(kind="bar",figsize=(20,4))
```

```
NaN                3
LOP110101000001    1
LOP150601001147    1
LOP150601001135    1
LOP150601001136    1
LOP150601001138    1
LOP150601001140    1
LOP150601001141    1
LOP150601001143    1
LOP150601001145    1
Name: Event Number, dtype: int64

<AxesSubplot:>
```



3. 数据缺失的处理

统计所有属性的数据缺失情况：

```
print(data.isnull().sum(axis=0))
```

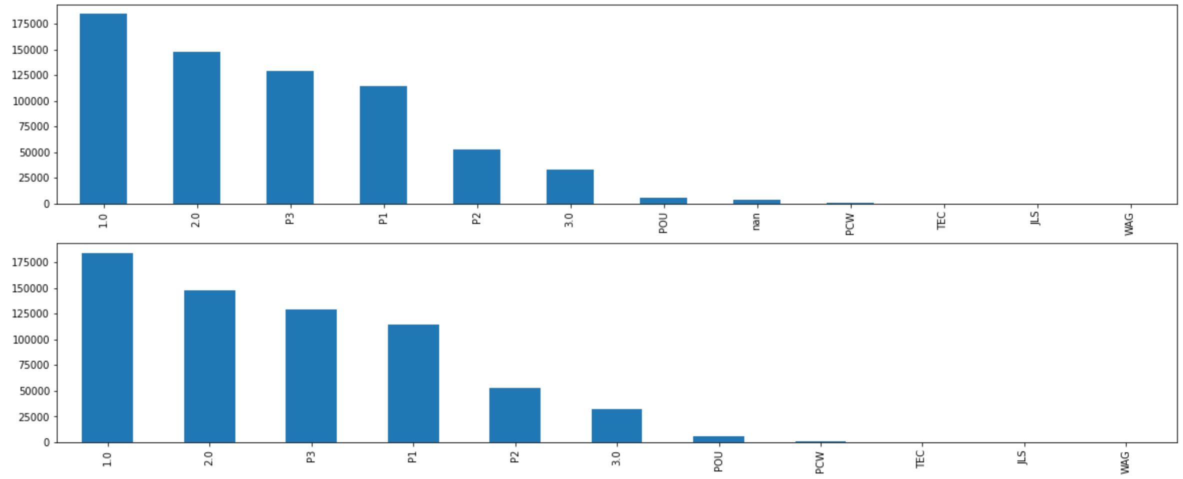
```
Agency                         3
Create Time                    3
Location                       0
Area Id                     3163
Beat                        3604
Priority                       3
Incident Type Id               3
Incident Type Description    250
Event Number                   3
Closed Time                   10
dtype: int64
```

3.1 处理Area Id属性缺失

缺失原因：统计失误，将缺失部分剔除

```
data_beat = data.dropna(subset=['Area Id'])
plt.subplot(2,1,1)
data["Area Id"].value_counts(dropna = False)[:50].plot(kind='bar',figsize=(20,8))
plt.subplot(2,1,2)
data_beat["Area Id"].value_counts(dropna = False)[:50].plot(kind='bar',figsize=(20,8))
```

```
<AxesSubplot:>
```

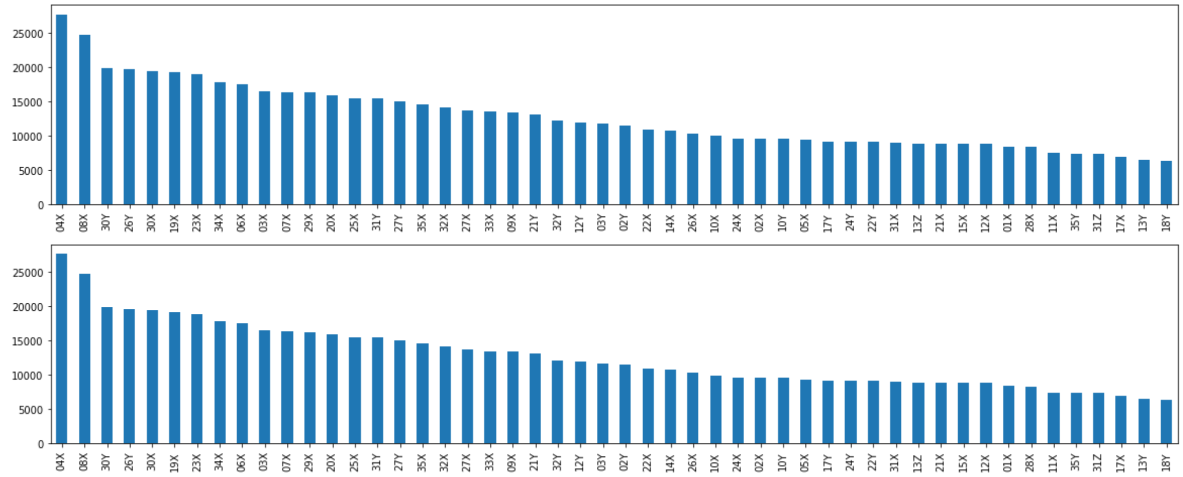


3.2 处理Beat属性缺失

缺失原因：统计失误，将缺失部分剔除

```
data_beat = data.dropna(subset=['Beat'])
plt.subplot(2,1,1)
data["Beat"].value_counts(dropna = False)[:50].plot(kind='bar',figsize=
(20,8))
plt.subplot(2,1,2)
data_beat["Beat"].value_counts(dropna = False)[:50].plot(kind='bar',figsize=
(20,8))
```

```
<AxesSubplot:>
```


25000
20000
15000
10000
5000
0
04X 08X 30Y 26Y 30X 19X 23X 34X 06X 03X 07X 29X 20X 25X 31Y 27Y 35X 32X 27X 33X 09X 21Y 32Y 12Y 03Y 02Y 22X 14X 26X 10X 24X 02X 10Y 05X 17Y 24Y 22Y 31X 13Z 21X 15X 12X 01X 28X 11X 35Y 31Z 17X 13Y 18Y
25000
20000
15000
10000
5000
0
04X 08X 30Y 26Y 30X 19X 23X 34X 06X 03X 07X 29X 20X 25X 31Y 27Y 35X 32X 27X 33X 09X 21Y 32Y 12Y 03Y 02Y 22X 14X 26X 10X 24X 02X 10Y 05X 17Y 24Y 22Y 31X 13Z 21X 15X 12X 01X 28X 11X 35Y 31Z 17X 13Y 18Y