2.Chicago Building Violations

github：https://github.com/AdamFocus/kaggleDataAnalysis (https://github.com/AdamFocus/kaggleDataAnalysis)

读取数据集

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
data = pd.read_csv('building-violations.csv', delimiter=',')
nRow, nCol = data.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
There are 1677788 rows and 32 columns
```

将时间属性格式转换

In [3]:

```
# dateparse = lambda dates: pd.datetime.strptime(dates, '%Y-%m-%dT%H')
data['VIOLATION DATE']=pd.to_datetime(data['VIOLATION DATE'],format='%Y-%m-%dT%H:%M:%S')
data['VIOLATION LAST MODIFIED DATE']=pd.to_datetime(data['VIOLATION LAST MODIFIED DATE'],format=
'%Y-%m-%dT%H:%M:%S')
data['VIOLATION STATUS DATE']=pd.to_datetime(data['VIOLATION STATUS DATE'],format='%Y-%m-%dT%H:%
M:%S')
```

数据可视化和摘要

标称属性

1.ID

In [4]:

```
IDvc=data.ID.value_counts()
IDvc
```

Out[4]:

```
5748742    2
5065831    2
2275291    2
2279535    2
5497931    2
          ..
6172316    1
6170269    1
6174367    1
2088610    1
4194304    1
Name: ID, Length: 1671242, dtype: int64
```

2.VIOLATION CODE

In [5]:

```
viocodeVC=data['VIOLATION CODE'].value_counts()
viocodeVC
```

Out[5]:

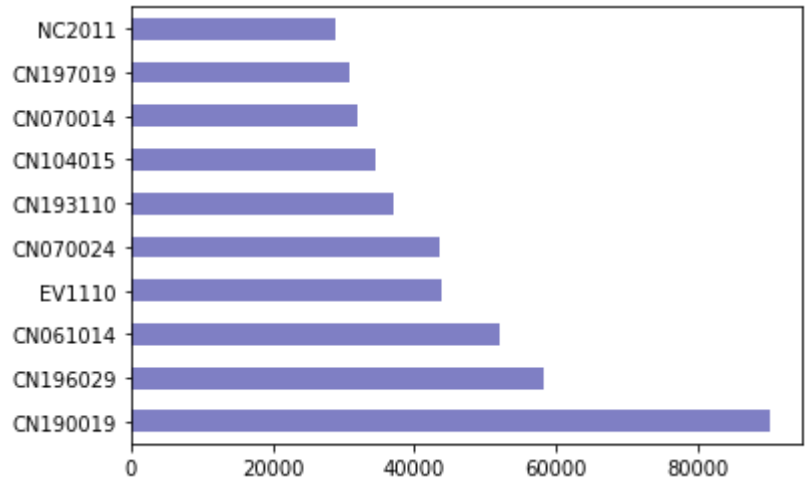
```
CN190019    89995
CN196029    58136
CN061014    51946
EV1110      43700
CN070024    43673
            ...
PL234034        1
RF309010        1
RF302020        1
ELM1437         1
VAP2503         1
Name: VIOLATION CODE, Length: 1468, dtype: int64
```

In [6]:

```
viocodetop10=viocodeVC.head(10)
viocodetop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f769230c8>
```



3.VIOLATION STATUS

In [7]:

```
viostaVC=data['VIOLATION STATUS'].value_counts()
viostaVC
```

Out[7]:

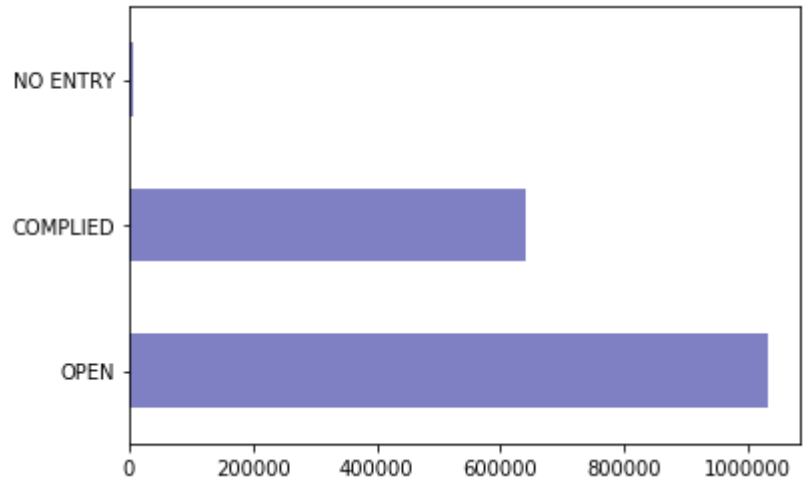
```
OPEN         1030958
COMPLIED      641247
NO ENTRY        5583
Name: VIOLATION STATUS, dtype: int64
```

In [8]:

```
viostaVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[8]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f59ec90c8>



4.VIOLATION DESCRIPTION

In [9]:

```
viodesVC=data['VIOLATION DESCRIPTION'].value_counts()
viodesVC
```

Out[9]:

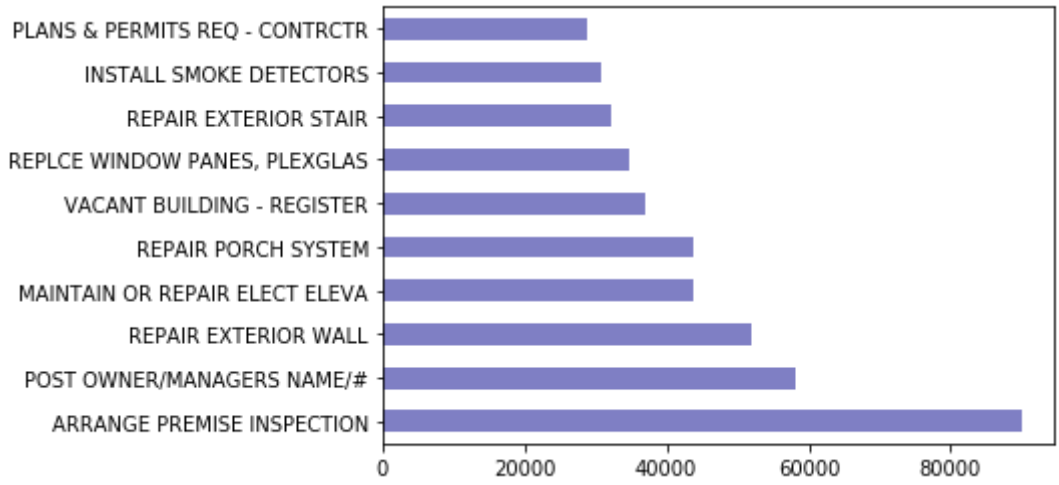
```
ARRANGE PREMISE INSPECTION        90004
POST OWNER/MANAGERS NAME/#        58136
REPAIR EXTERIOR WALL              51946
MAINTAIN OR REPAIR ELECT ELEVA    43700
REPAIR PORCH SYSTEM               43673
                                  ...
OBTAIN CANOPY PERMIT                  1
REPL BRAKE LINING FRT                 1
14E-7-701.8:GENERATOR LOCATION        1
PROVIDE APPR FIRE STOPPING.           1
PROVIDE CALCS: TRAVEL DISTANCE        1
Name: VIOLATION DESCRIPTION, Length: 1312, dtype: int64
```

In [10]:

```
viodesVCtop10=viodesVC.head(10)
viodesVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[10]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f6486aac8>



5.VIOLATION LOCATION

In [11]:

```
violocVC=data['VIOLATION LOCATION'].value_counts()
violocVC
```

Out[11]:

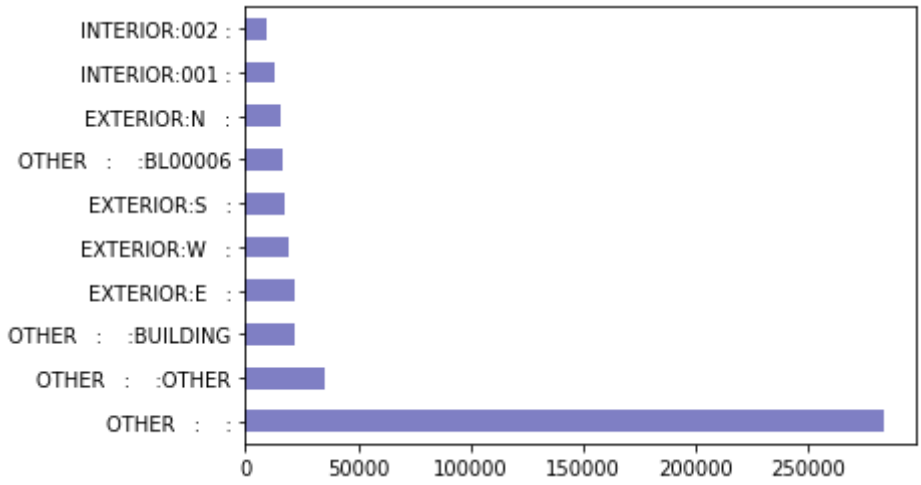
```
OTHER    :     :                                                284277
OTHER    :     :OTHER                                            35182
OTHER    :     :BUILDING                                         21934
EXTERIOR:E    :                                                 21522
EXTERIOR:W    :                                                 19460
                                                                  ...
EXTERIOR:W    :1ST FLOOR - NORTH:                                   1
2 STAIRWELLS(N/S)                                                   1
EXTERIOR:W    :WEST AT 4TH 2ND, EAST AT 3RD                         1
EXTERIOR:N    :NORTH, SOUTH ELEVATIONS; INTERIOR STAIRWELL          1
REAR UNIT                                                           1
Name: VIOLATION LOCATION, Length: 52821, dtype: int64
```

In [12]:

```
violocVCtop10=violocVC.head(10)
violocVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[12]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f6b268408>



6.VIOLATION INSPECTOR COMMENTS

In [13]:

```
vioicVC=data['VIOLATION INSPECTOR COMMENTS'].value_counts()
vioicVC
```

Out[13]:

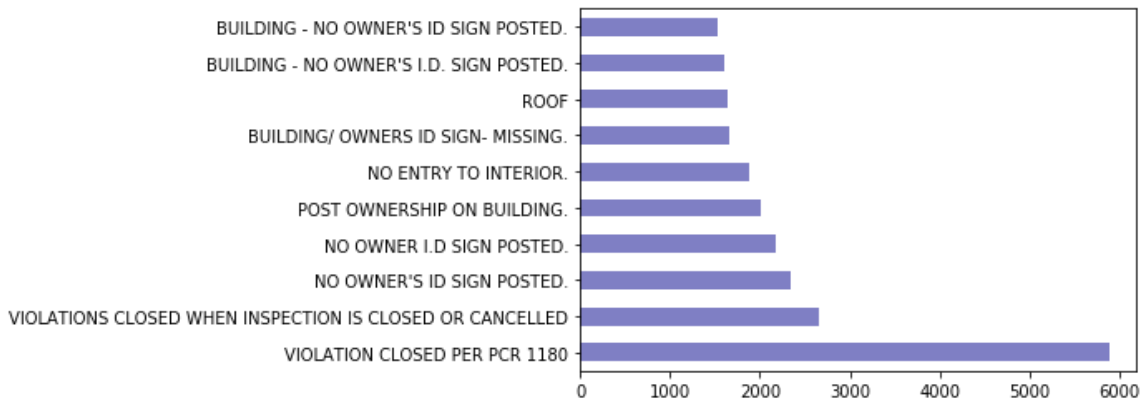
```
VIOLATION CLOSED PER PCR 1180
5889
VIOLATIONS CLOSED WHEN INSPECTION IS CLOSED OR CANCELLED
2650
NO OWNER'S ID SIGN POSTED.
2347
NO OWNER I.D SIGN POSTED.
2184
POST OWNERSHIP ON BUILDING.
2006


...
SOUTHWEST DOOR OLD BUILDING EXIT SIGN MISSING.
1
REAR ,UNDER PORCH STAIRS , JUNK AND DEBRIS PILED .
1
REAR YARD JUNK AND DEBRIS, BROKEN UP CONCRETE WALK AT BUILDING FRONT
1
EAST BALCONY SOFFIT FLAKY PAINT
1
EAST AND NORTH ELEVATIONS/ VARIOUS WINDOW SASHES - DRY-ROTTED WITH PEELING PAINT.
1
Name: VIOLATION INSPECTOR COMMENTS, Length: 1053818, dtype: int64
```

In [14]:

```
vioicVCtop10=vioicVC.head(10)
vioicVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f67f4f648>
```



7.VIOLATION ORDINANCE

In [15]:

```
viooVC=data['VIOLATION ORDINANCE'].value_counts()
viooVC
```

Out[15]:

```
Arrange for inspection of premises. (13-12-100)
89995
Post name, address, and telephone of owner, owner's agent for managing, controllin
g or collecting rents, and any other person managing or controlling building consp
icuously where accessible or visible to public way. (13-12-030)    58136
Failed to maintain the exterior walls of a building or structure free from holes,
breaks, loose or rotting boards or timbers and any other conditions which might ad
mit rain or dampness to the walls.  (13-196-530(b), 13-196-641)    51946
Failed to maintain electric elevator equipment provided at premises in safe and so
und working condition. (13-196-590, 13-196-630(b), 18-30-001)
43700
Failed to repair or replace defective or missing members of porch system.  (13-196
-570, 13-196-641)
43673

...
Arrange for reinspection. (13-12-100)
1
Repair or replace defective main floor fire service key switch for freight elevato
r. (13-156-460 1, A, B, C, D, 13-20-120)
1
Repair defective belt tension for man-lift. (13-156-010, 13-20-120, ANSI A90.1-196
9)
1
Failure To Carry A Valid City Of Chicago Crane License Or Apprentice Certificate W
hile Operating A Crane. 1(4-288-010) 1(4-288-120)
1
Repair or replace defective horizontally sliding hoistway door hangars for passeng
er elevator. (13-156-010, 13-20-120, ANSI A17.1-1971, rule 110.11 D)
1
Name: VIOLATION ORDINANCE, Length: 1306, dtype: int64
```

In [16]:

```
viooVCtop10=viooVC.head(10)
viooVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f55e32988>
```

8.INSPECTOR ID

In [17]:

```
iIDVC=data['INSPECTOR ID'].value_counts()
iIDVC
```

Out[17]:

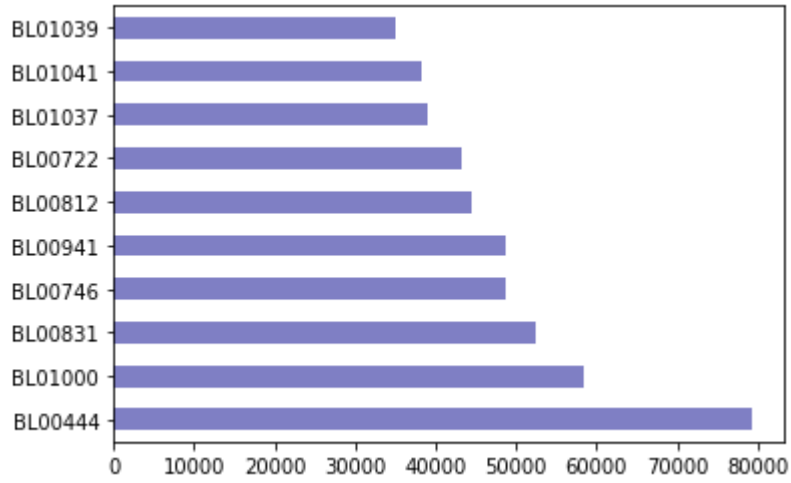
```
BL00444          79336
BL01000          58497
BL00831          52450
BL00746          48797
BL00941          48717
                 ...
BL00888              2
CN00085PL            2
TESTEMPLOYEE         2
DS00009              1
557556               1
Name: INSPECTOR ID, Length: 361, dtype: int64
```

In [18]:

```
iIDVCtop10=iIDVC.head(10)
iIDVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f560e7208>
```



9.INSPECTION STATUS

In [19]:

```
istaVC=data['INSPECTION STATUS'].value_counts()
istaVC
```

Out[19]:

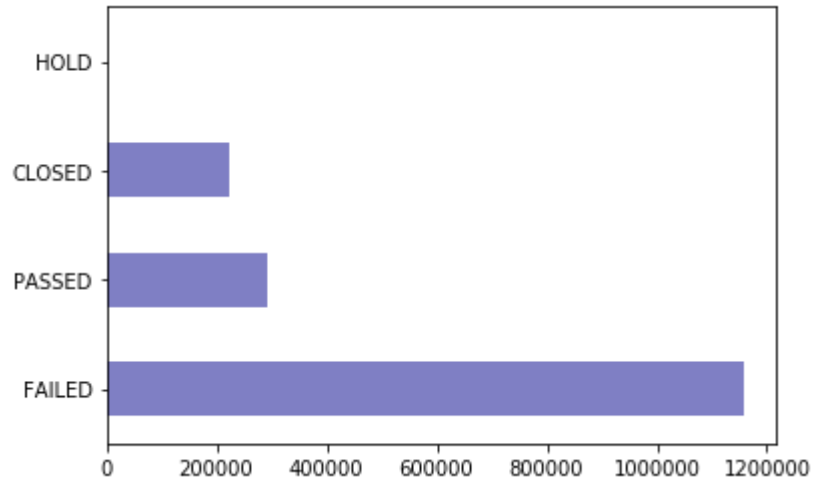
```
FAILED     1159758
PASSED      293076
CLOSED      224784
HOLD           154
Name: INSPECTION STATUS, dtype: int64
```

In [20]:

```
istaVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[20]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f6ea1c388>



10.INSPECTION WAIVED

In [21]:

```
iwVC=data['INSPECTION WAIVED'].value_counts()
iwVC
```

Out[21]:

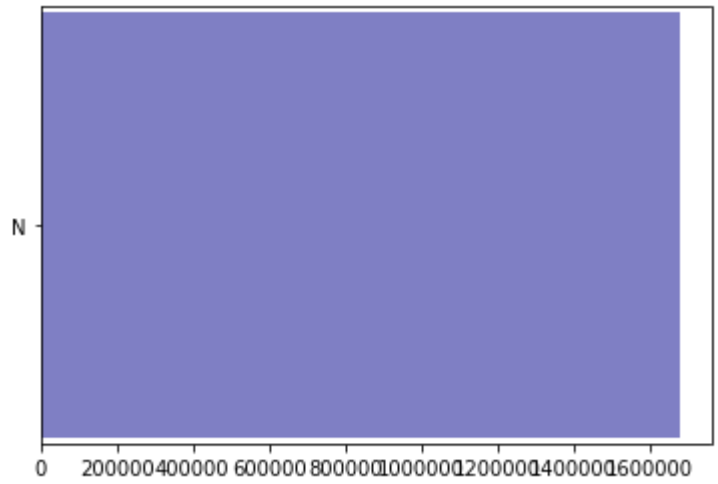
```
N     1677788
Name: INSPECTION WAIVED, dtype: int64
```

In [22]:

```
iwVC.plot(kind='barh', color='darkblue', alpha=0.5,width=20)
```

Out[22]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f72de3548>



11.INSPECTION CATEGORY

In [23]:

```
icVC=data['INSPECTION CATEGORY'].value_counts()
icVC
```

Out[23]:

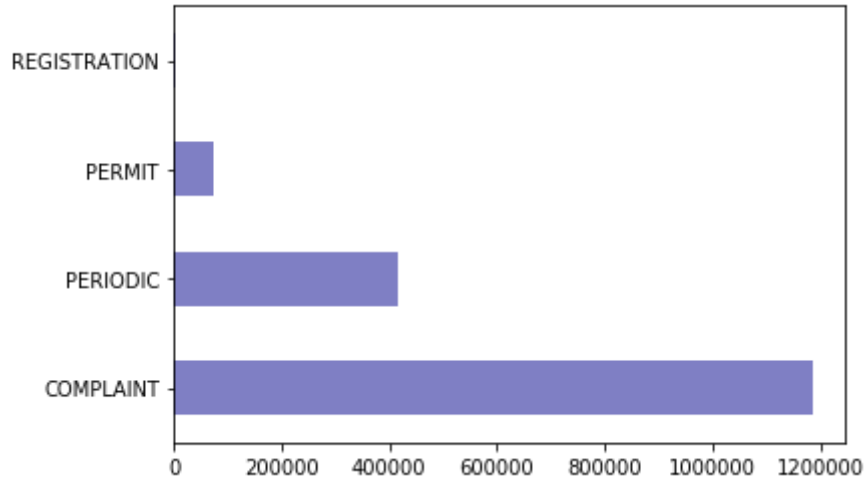
```
COMPLAINT       1186426
PERIODIC         415176
PERMIT            73600
REGISTRATION       2586
Name: INSPECTION CATEGORY, dtype: int64
```

In [24]:

```
icVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f6d396648>
```



12.DEPARTMENT BUREAU

In [25]:

```
dbVC=data['DEPARTMENT BUREAU'].value_counts()
dbVC
```

Out[25]:

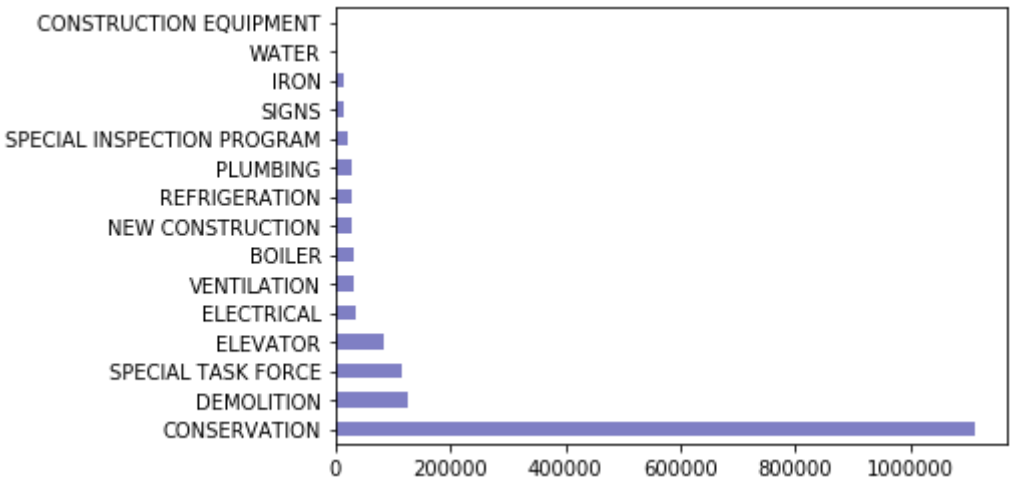
```
CONSERVATION                  1110911
DEMOLITION                     125464
SPECIAL TASK FORCE             115885
ELEVATOR                        85805
ELECTRICAL                      37243
VENTILATION                     32108
BOILER                          31235
NEW CONSTRUCTION                29938
REFRIGERATION                   29681
PLUMBING                        28199
SPECIAL INSPECTION PROGRAM      21033
SIGNS                           14539
IRON                            14405
WATER                             786
CONSTRUCTION EQUIPMENT            556
Name: DEPARTMENT BUREAU, dtype: int64
```

In [26]:

```
dbVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[26]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f641dcac8>
```



13.ADDRESS

In [27]:

```
addVC=data['ADDRESS'].value_counts()
addVC
```

Out[27]:

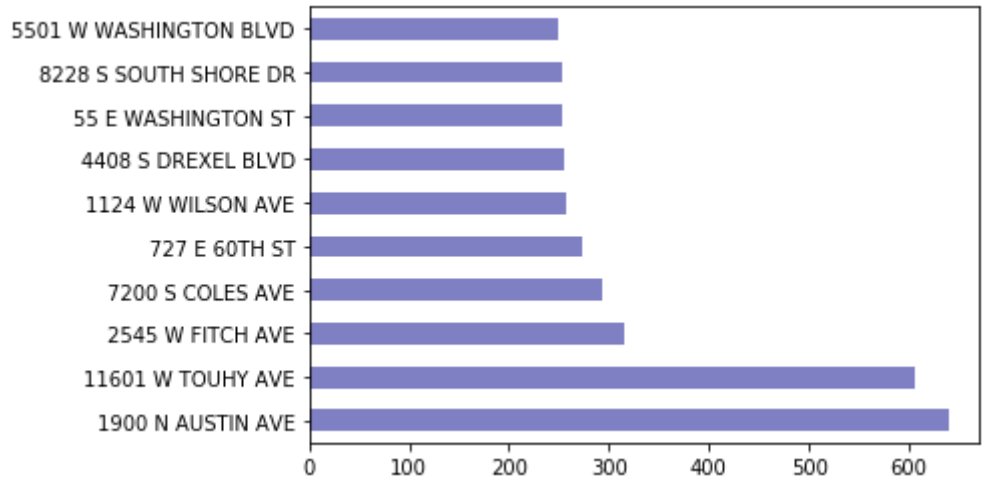
```
1900 N AUSTIN AVE     639
11601 W TOUHY AVE     605
2545 W FITCH AVE      316
7200 S COLES AVE      294
727 E 60TH ST         274
                     ...
2955 N TROY ST          1
1524 W JARVIS AVE       1
1429 W BARRY AVE        1
7322 S PERRY AVE        1
1712 W WARREN BLVD      1
Name: ADDRESS, Length: 154007, dtype: int64
```

In [28]:

```
addVCtop10=addVC.head(10)
addVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f748c1a08>
```



14.STREET DIRECTION

In [29]:

```
sdVC=data['STREET DIRECTION'].value_counts()
sdVC
```

Out[29]:

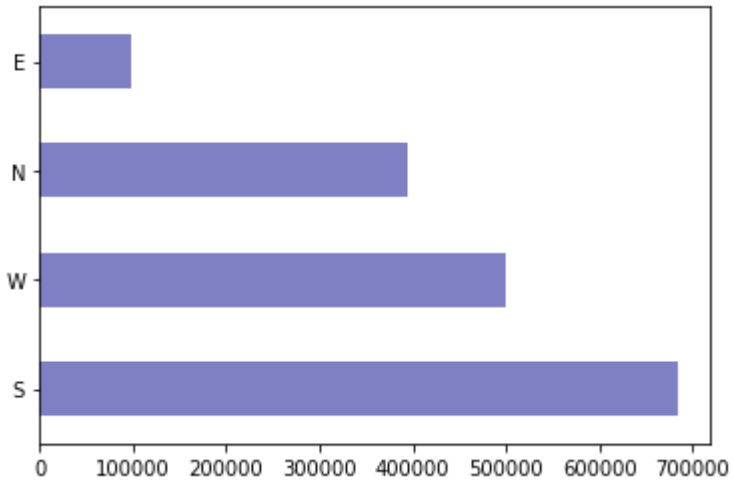
```
S     683917
W     500418
N     395246
E      98207
Name: STREET DIRECTION, dtype: int64
```

In [30]:

```
sdVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[30]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f678bda88>



15.STREET NAME

In [31]:

```
snVC=data['STREET NAME'].value_counts()
snVC
```

Out[31]:

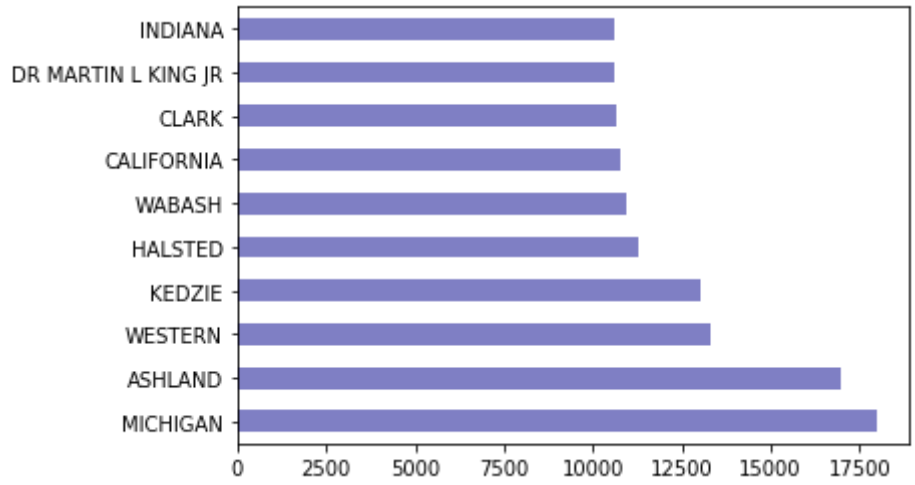
```
MICHIGAN      17980
ASHLAND       16992
WESTERN       13327
KEDZIE        13034
HALSTED       11268
              ...
GRADY             1
LIVERMORE         1
LONDON            1
CRILLY            1
MELODY            1
Name: STREET NAME, Length: 1190, dtype: int64
```

In [32]:

```
snVCtop10=snVC.head(10)
snVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[32]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f57aaf608>



16.STREET TYPE

In [33]:

```
stVC=data['STREET TYPE'].value_counts()
stVC
```

Out[33]:

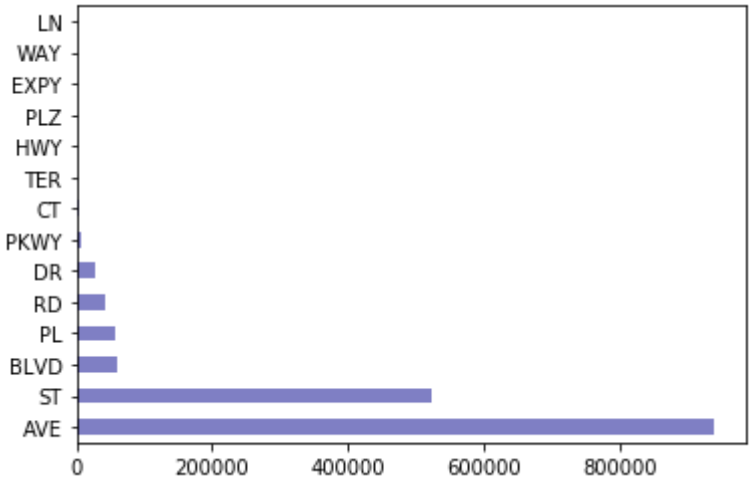
```
AVE     940725
ST      523743
BLVD     59536
PL       57665
RD       41100
DR       27145
PKWY      6605
CT        3287
TER       2222
HWY       1559
PLZ        639
EXPY        10
WAY          8
LN           3
Name: STREET TYPE, dtype: int64
```

In [34]:

```
stVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[34]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f665b0c88>



17.PROPERTY GROUP

In [35]:

```
pgVC=data['PROPERTY GROUP'].value_counts()
pgVC
```

Out[35]:

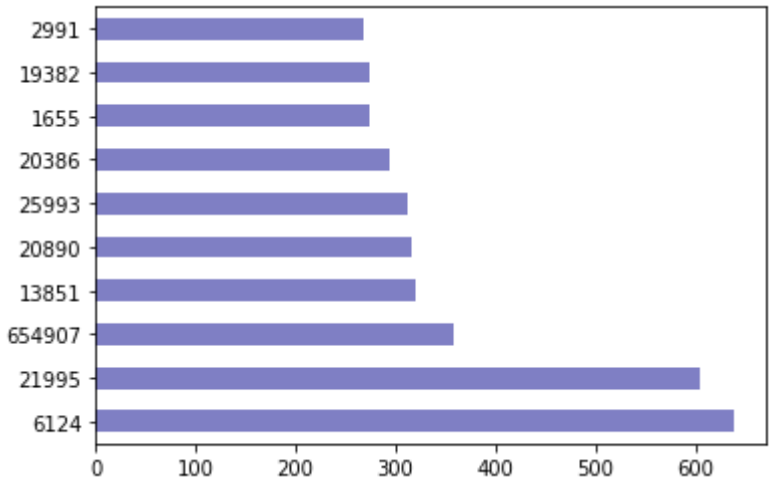
```
6124      639
21995     605
654907    359
13851     320
20890     316
         ...
63958       1
505636      1
284714      1
292910      1
148236      1
Name: PROPERTY GROUP, Length: 138301, dtype: int64
```

In [36]:

```
pgVCtop10=pgVC.head(10)
pgVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[36]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f5ceed788>



18.LOCATION

In [37]:

```
locVC=data['LOCATION'].value_counts()
locVC
```

Out[37]:

```
{'latitude': '41.91482042670598', 'human_address': '{"address": "", "city": "", "s
tate": "", "zip": ""}', 'longitude': '-87.77555966968384'}      639
{'latitude': '42.008536400868735', 'human_address': '{"address": "", "city": "",
"state": "", "zip": ""}', 'longitude': '-87.91442843927047'}     605
{'latitude': '42.011175414582205', 'human_address': '{"address": "", "city": "",
"state": "", "zip": ""}', 'longitude': '-87.69436142514081'}     316
{'latitude': '41.76505512958016', 'human_address': '{"address": "", "city": "", "s
tate": "", "zip": ""}', 'longitude': '-87.5636984239516'}       294
{'latitude': '41.78574174363065', 'human_address': '{"address": "", "city": "", "s
tate": "", "zip": ""}', 'longitude': '-87.60740121841071'}      274

...
{'latitude': '41.88231683582112', 'human_address': '{"address": "", "city": "", "s
tate": "", "zip": ""}', 'longitude': '-87.66356706707126'}        1
{'latitude': '41.775777664342925', 'human_address': '{"address": "", "city": "",
"state": "", "zip": ""}', 'longitude': '-87.6156612545782'}       1
{'latitude': '41.75208284503481', 'human_address': '{"address": "", "city": "", "s
tate": "", "zip": ""}', 'longitude': '-87.66228827432326'}        1
{'latitude': '41.651939202825716', 'human_address': '{"address": "", "city": "",
"state": "", "zip": ""}', 'longitude': '-87.6150094733589'}       1
{'latitude': '41.932504081782085', 'human_address': '{"address": "", "city": "",
"state": "", "zip": ""}', 'longitude': '-87.6691623397685'}       1
Name: LOCATION, Length: 153766, dtype: int64
```

In [38]:

```
locVCtop10=locVC.head(10)
locVCtop10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[38]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f59c20948>
```

19.VIOLATION LAST MODIFIED DATE

In [39]:

```
vlmdateVC=data["VIOLATION LAST MODIFIED DATE"].value_counts()
vlmdateVC
```

Out[39]:

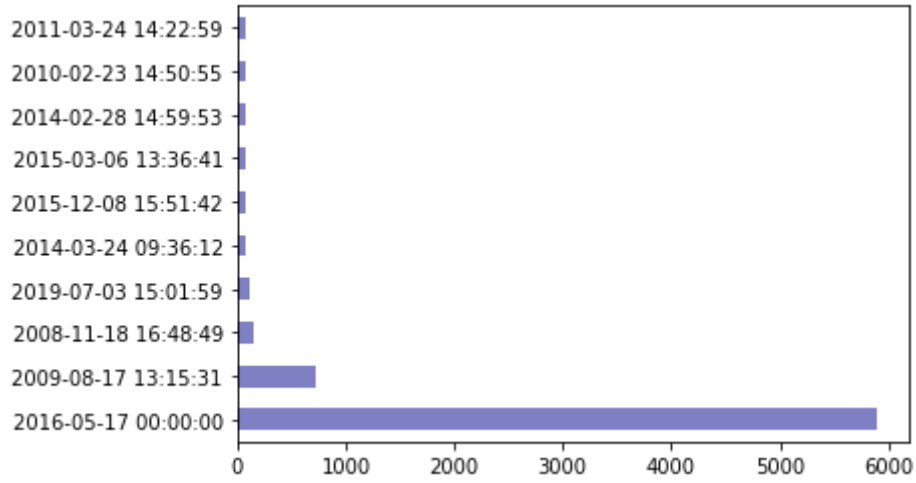
```
2016-05-17 00:00:00    5889
2009-08-17 13:15:31     717
2008-11-18 16:48:49     156
2019-07-03 15:01:59     116
2014-03-24 09:36:12      82
                        ...
2007-12-03 15:04:39       1
2008-07-09 13:30:30       1
2012-05-29 11:01:20       1
2018-02-05 11:38:45       1
2006-05-01 11:37:31       1
Name: VIOLATION LAST MODIFIED DATE, Length: 953167, dtype: int64
```

In [40]:

```
vlmdateVC10=vlmdateVC.head(10)
vlmdateVC10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[40]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f7810d2c8>
```



20.VIOLATION DATE

In [41]:

```
vdateVC=data["VIOLATION DATE"].value_counts()
vdateVC
```

Out[41]:

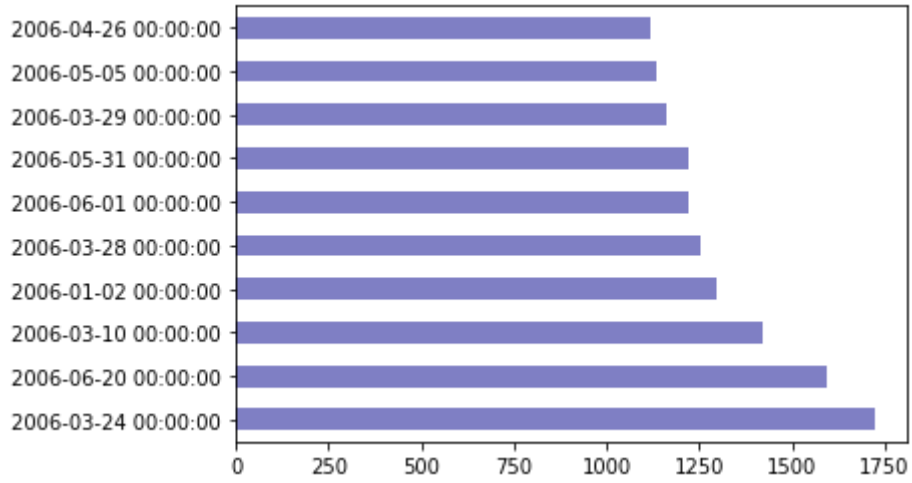
```
2006-03-24    1723
2006-06-20    1595
2006-03-10    1423
2006-01-02    1299
2006-03-28    1252
              ...
2019-04-28       1
2017-03-12       1
2011-08-27       1
2016-03-13       1
2013-01-05       1
Name: VIOLATION DATE, Length: 4463, dtype: int64
```

In [42]:

```
vdateVC10=vdateVC.head(10)
vdateVC10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f73e957c8>
```



21.VIOLATION STATUS DATE

In [43]:

```
vsdateVC=data['VIOLATION STATUS DATE'].value_counts()
vsdateVC
```

Out[43]:

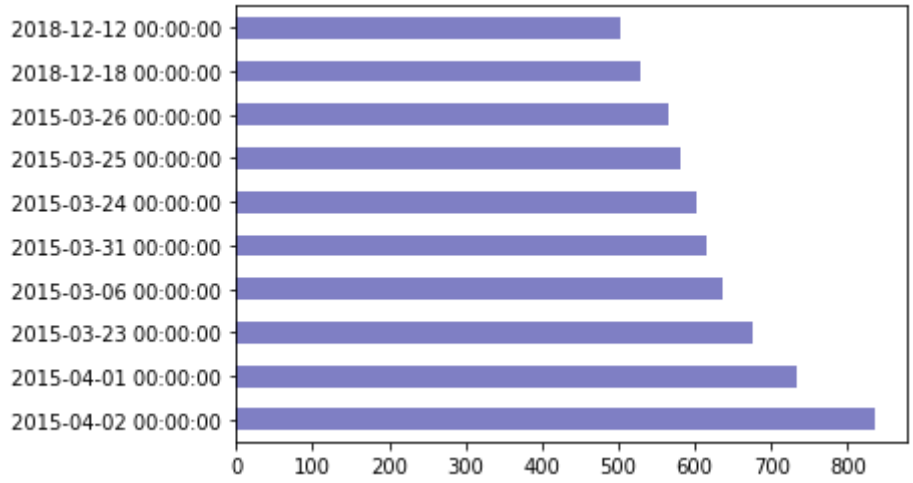
```
2015-04-02    836
2015-04-01    734
2015-03-23    675
2015-03-06    636
2015-03-31    615
              ...
2009-06-20      1
2007-12-01      1
2011-05-29      1
2010-10-23      1
2016-05-29      1
Name: VIOLATION STATUS DATE, Length: 4183, dtype: int64
```

In [44]:

```
vsdateVC10=vsdateVC.head(10)
vsdateVC10.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[44]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26f7bef9a48>
```



数值属性

In [45]:

```
def fiveNumber(name):
    #五数概括 Minimum（最小值）、Q1、Median（中位数、）、Q3、Maximum（最大值）
    Minimum=data[name].min()
    Maximum=data[name].max()
    Q1 = data[name].describe()['25%']
    Q3 = data[name].describe()['75%']
    Median=data[name].describe()['50%']
    IQR = Q3-Q1
    lower_limit=Q1-1.5*IQR #下限值
    upper_limit=Q3+1.5*IQR #上限值
    return [Minimum,Q1,Median,Q3,Maximum,lower_limit,upper_limit]
def boxplot(name):
    fig,axes=plt.subplots()
    data[name].plot(kind='box',ax=axes)
    axes.set_ylabel(name)
    fig.savefig(name+'.png')
def hist(name):
    plt.figure(figsize = (8, 6))
    plt.hist(data[name].dropna(), bins = 50, edgecolor = 'black')
    plt.xlabel(name)
    plt.ylabel('Count')
```

22.INSPECTION NUMBER

In [46]:

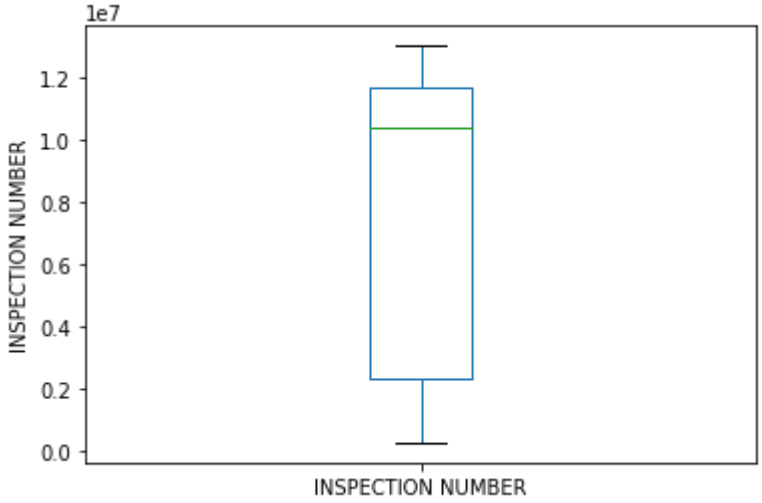
```
print('INSPECTION NUMBER五数概括:',fiveNumber('INSPECTION NUMBER'))
```

```
INSPECTION NUMBER五数概括: [265575, 2304416.0, 10418746.0, 11687280.0, 13050915, -11769880.0, 25761576.0]
```

盒图

In [47]:

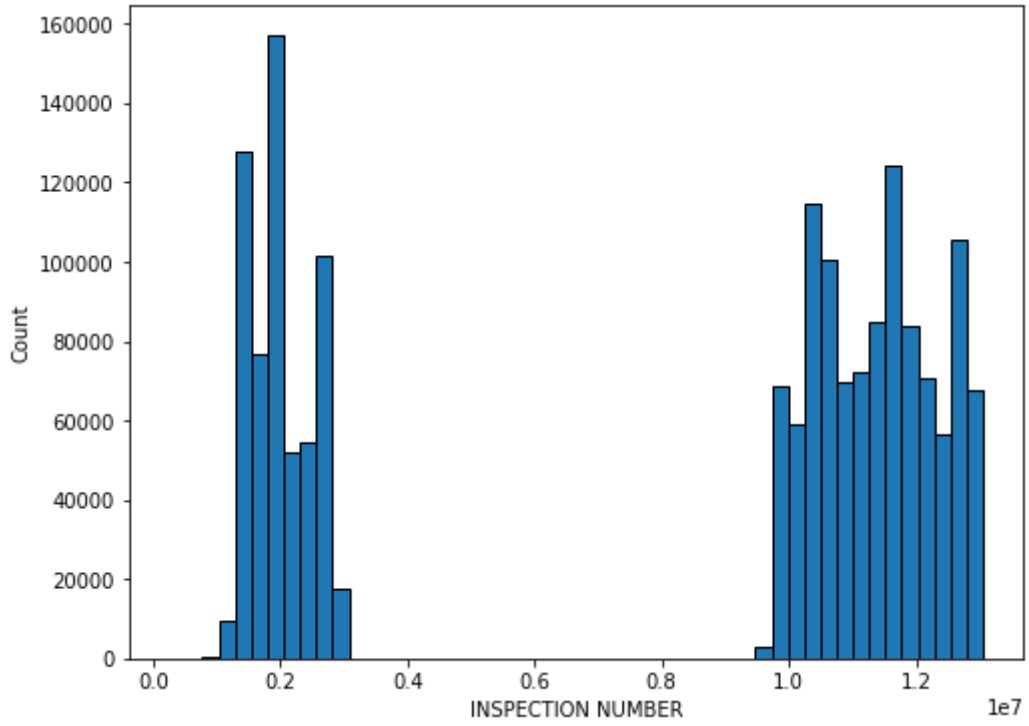
```
boxplot('INSPECTION NUMBER')
```



直方图

In [48]:

```
hist('INSPECTION NUMBER')
```



23.STREET NUMBER

In [49]:

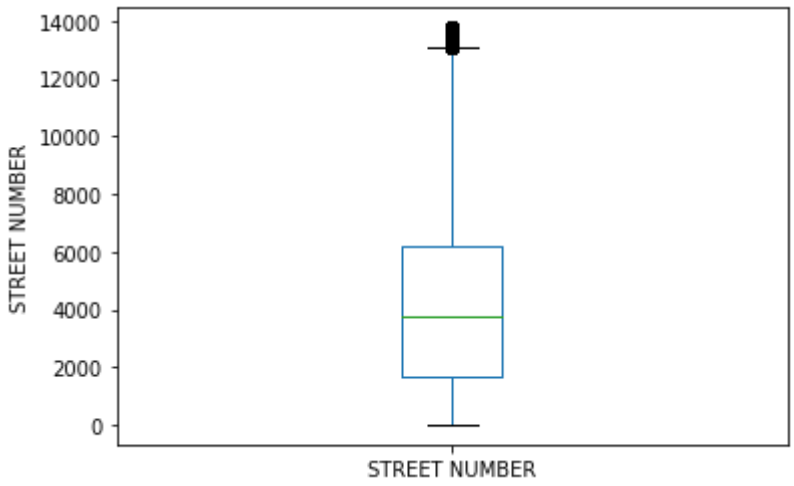
```
print('STREET NUMBER五数概括:',fiveNumber('STREET NUMBER'))
```

STREET NUMBER五数概括: [1, 1648.0, 3747.0, 6228.0, 13770, -5222.0, 13098.0]

盒图

In [50]:

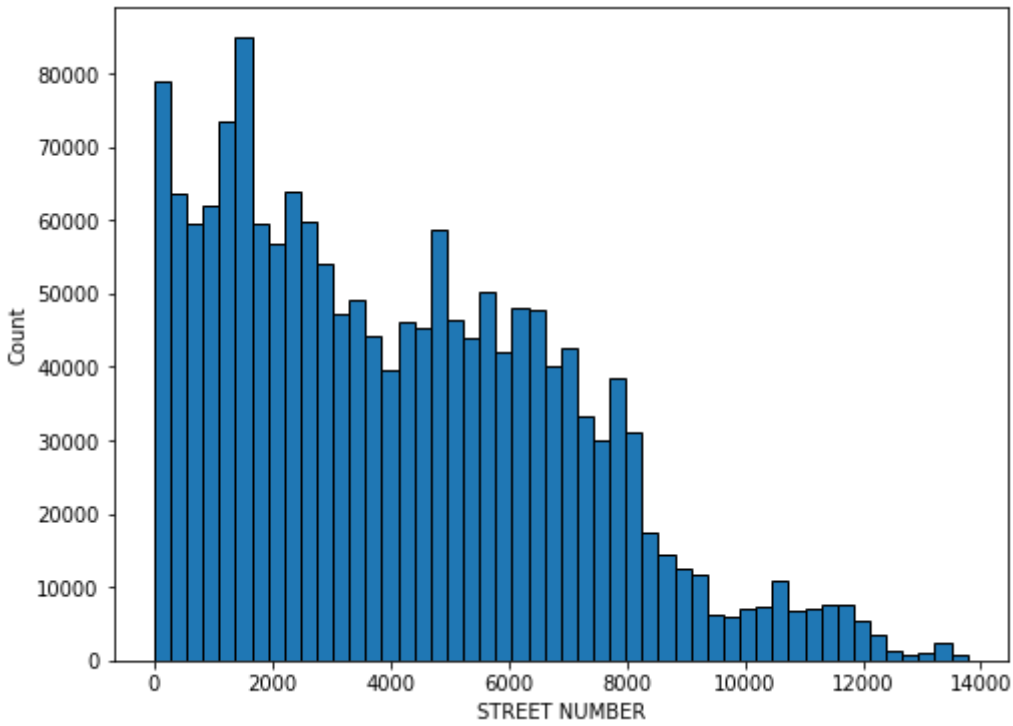
```
boxplot('STREET NUMBER')
```



直方图

In [51]:

```
hist('STREET NUMBER')
```



24.SSA

In [52]:

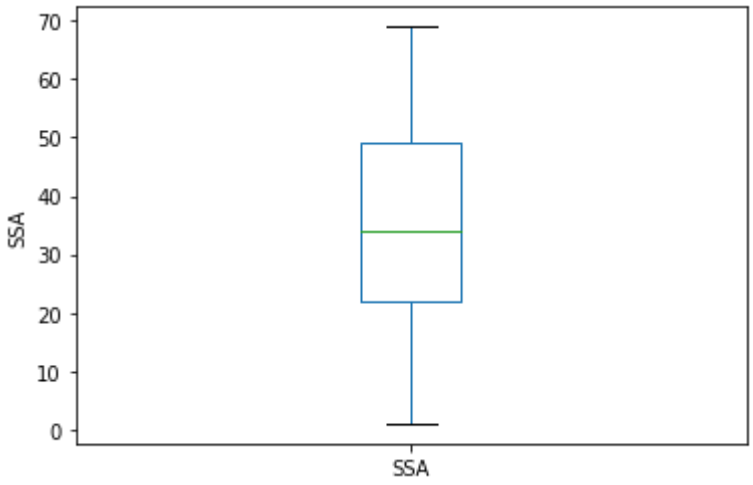
```
print('SSA五数概括:',fiveNumber('SSA'))
```

SSA五数概括: [1.0, 22.0, 34.0, 49.0, 69.0, -18.5, 89.5]

盒图

In [53]:

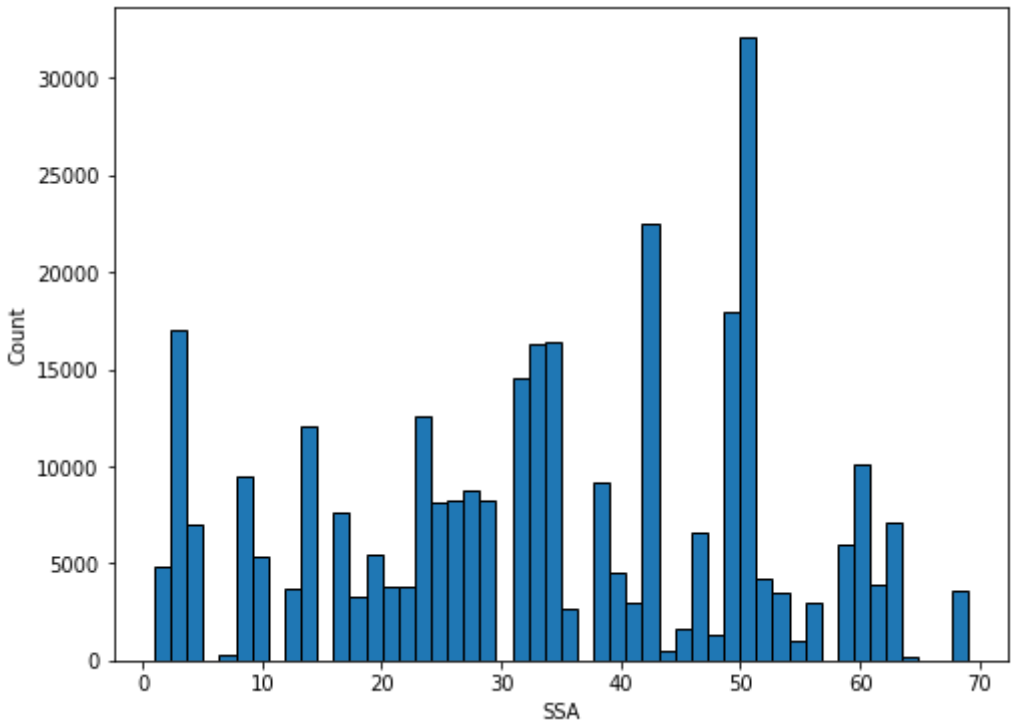
```
boxplot('SSA')
```



直方图

In [54]:

```
hist('SSA')
```



25.LATITUDE

In [55]:

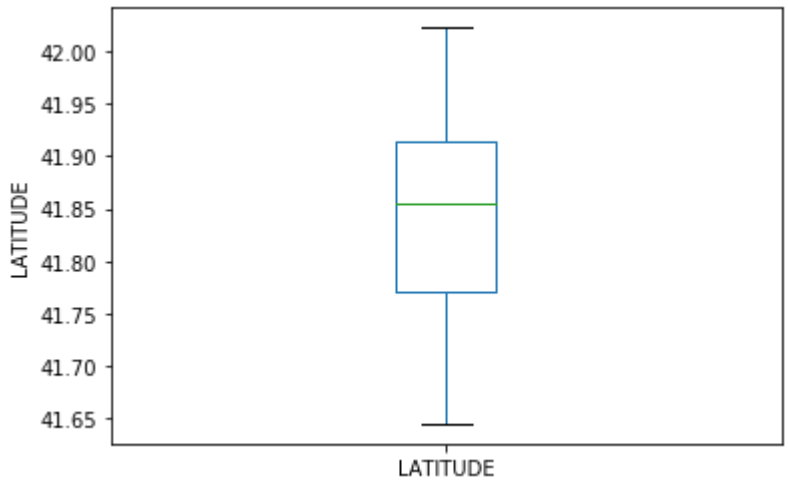
```
print('LATITUDE五数概括:',fiveNumber('LATITUDE'))
```

```
LATITUDE五数概括: [41.644670131999995, 41.770896504250004, 41.85400233599999, 41.9
13504192, 42.02268599, 41.55698497262502, 42.12741572362499]
```

盒图

In [56]:

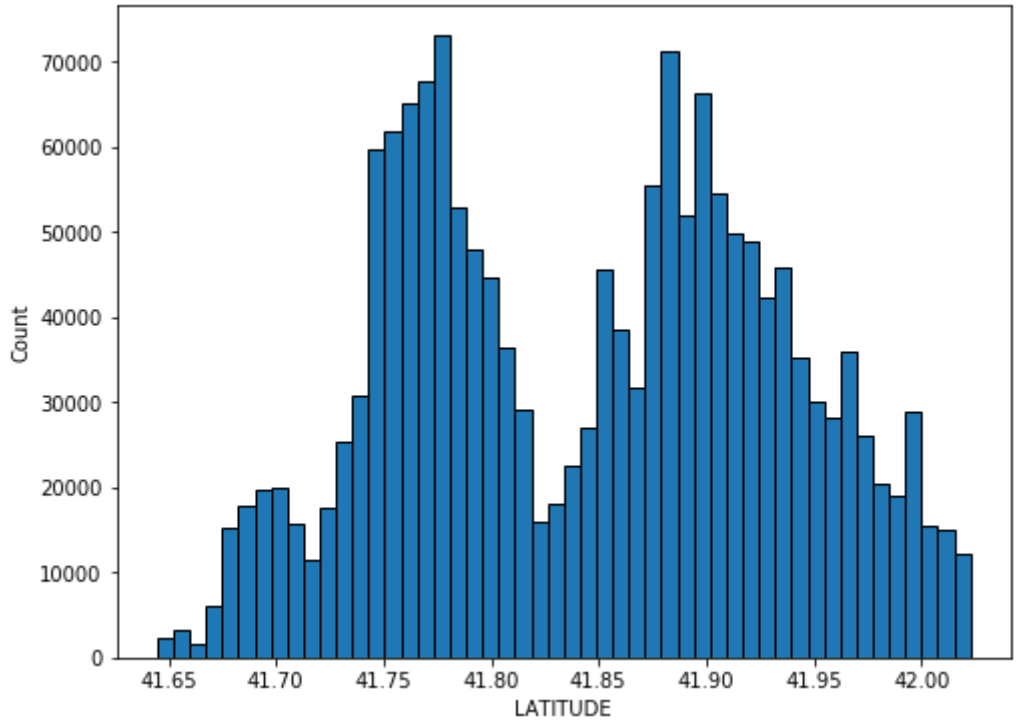
```
boxplot('LATITUDE')
```



直方图

In [57]:

```
hist('LATITUDE')
```



26.LONGITUDE

In [58]:

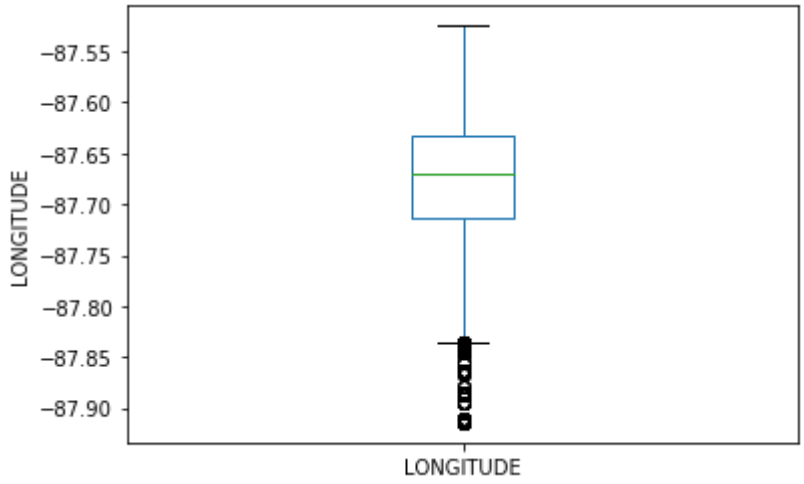
```
print('LONGITUDE五数概括:',fiveNumber('LONGITUDE'))
```

```
LONGITUDE五数概括: [-87.914435848, -87.71391769799999, -87.6698535045, -87.6328827
44, -87.524679151, -87.83547012899997, -87.51133031300002]
```

盒图

In [59]:

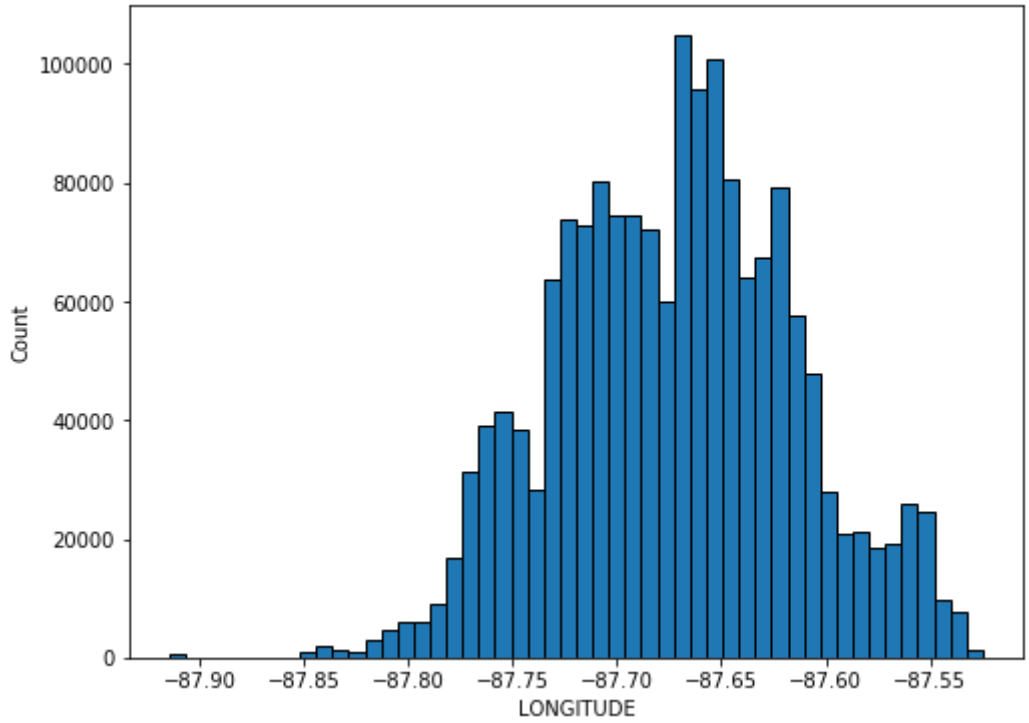
```
boxplot('LONGITUDE')
```



直方图

In [60]:

```
hist('LONGITUDE')
```



27.Community Areas

In [61]:

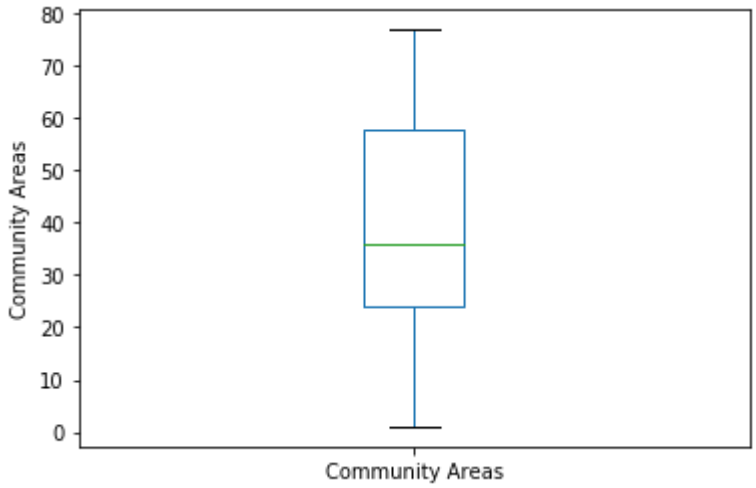
```
print('Community Areas五数概括:',fiveNumber('Community Areas'))
```

```
Community Areas五数概括: [1.0, 24.0, 36.0, 58.0, 77.0, -27.0, 109.0]
```

盒图

In [62]:

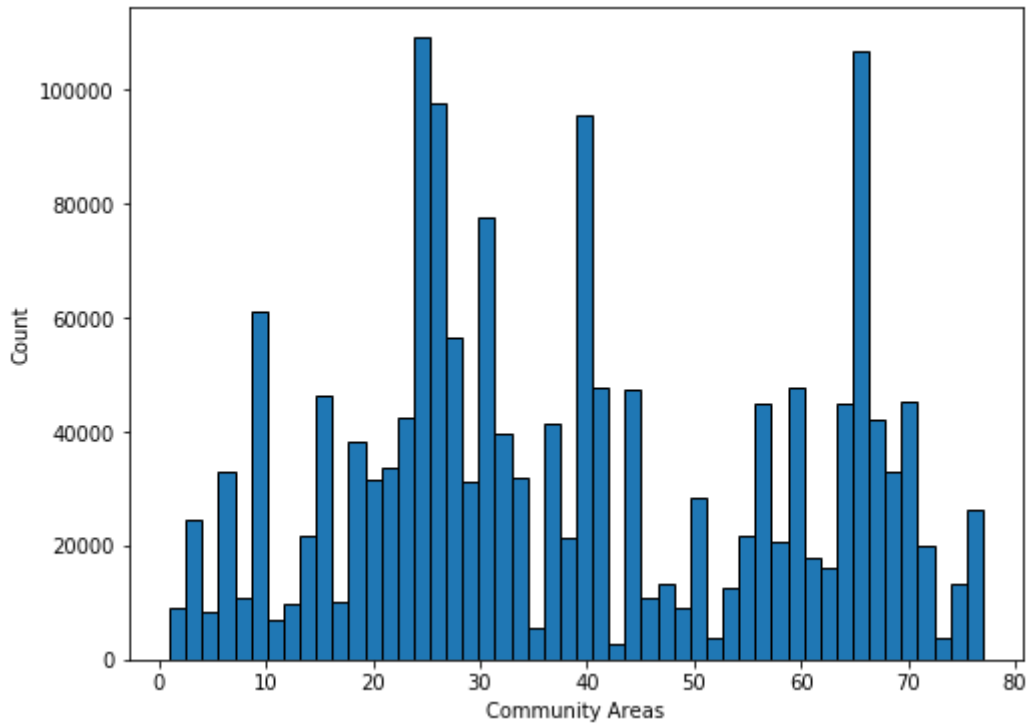
```
boxplot('Community Areas')
```



直方图

In [63]:

```
hist('Community Areas')
```



28.Zip Codes

In [64]:

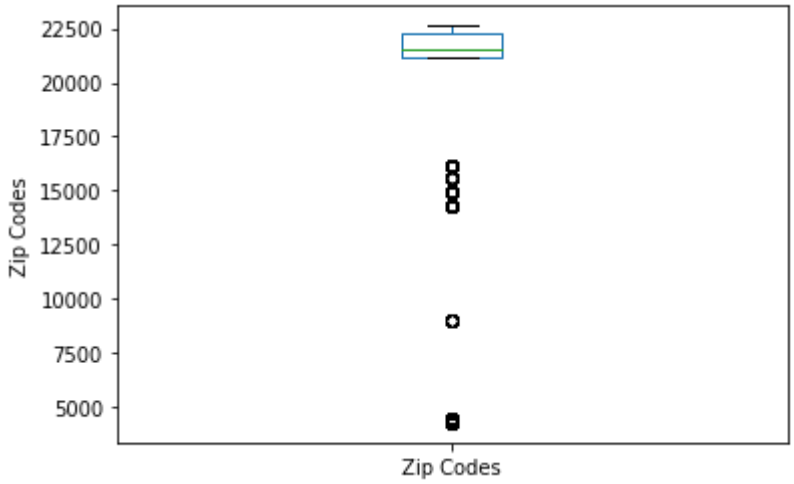
```
print('Zip Codes五数概括:',fiveNumber('Zip Codes'))
```

```
Zip Codes五数概括: [4299.0, 21190.0, 21569.0, 22248.0, 22620.0, 19603.0, 23835.0]
```

盒图

In [65]:

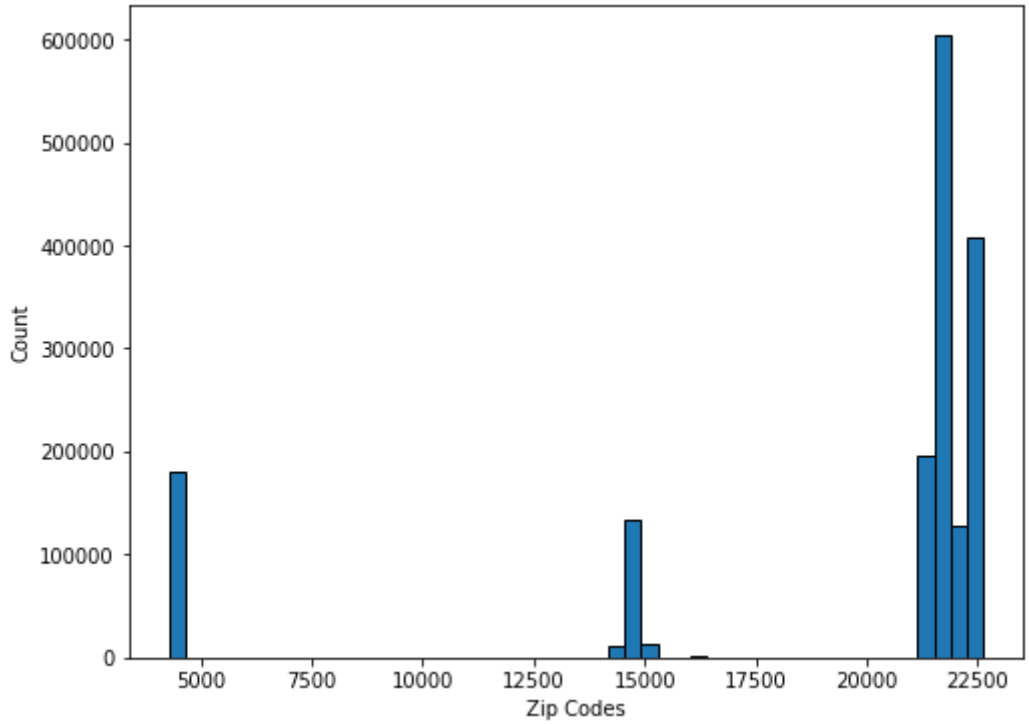
```
boxplot('Zip Codes')
```



直方图

In [66]:

```
hist('Zip Codes')
```



29.Boundaries - ZIP Codes

In [67]:

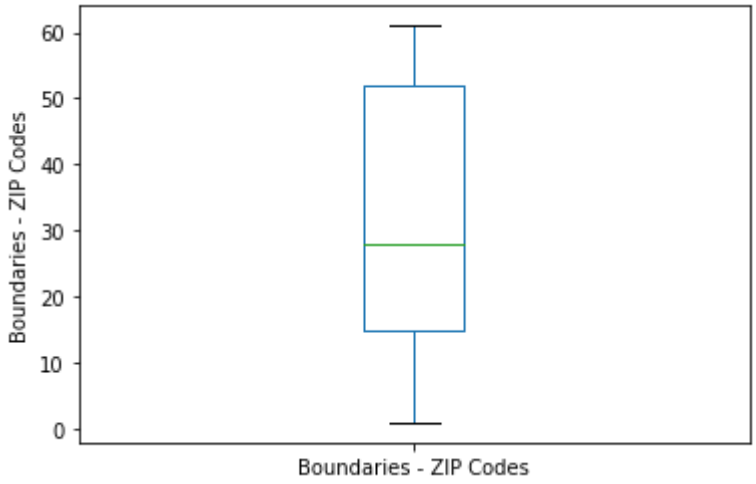
```
print('Boundaries - ZIP Codes五数概括:',fiveNumber('Boundaries - ZIP Codes'))
```

```
Boundaries - ZIP Codes五数概括: [1.0, 15.0, 28.0, 52.0, 61.0, -40.5, 107.5]
```

盒图

In [68]:

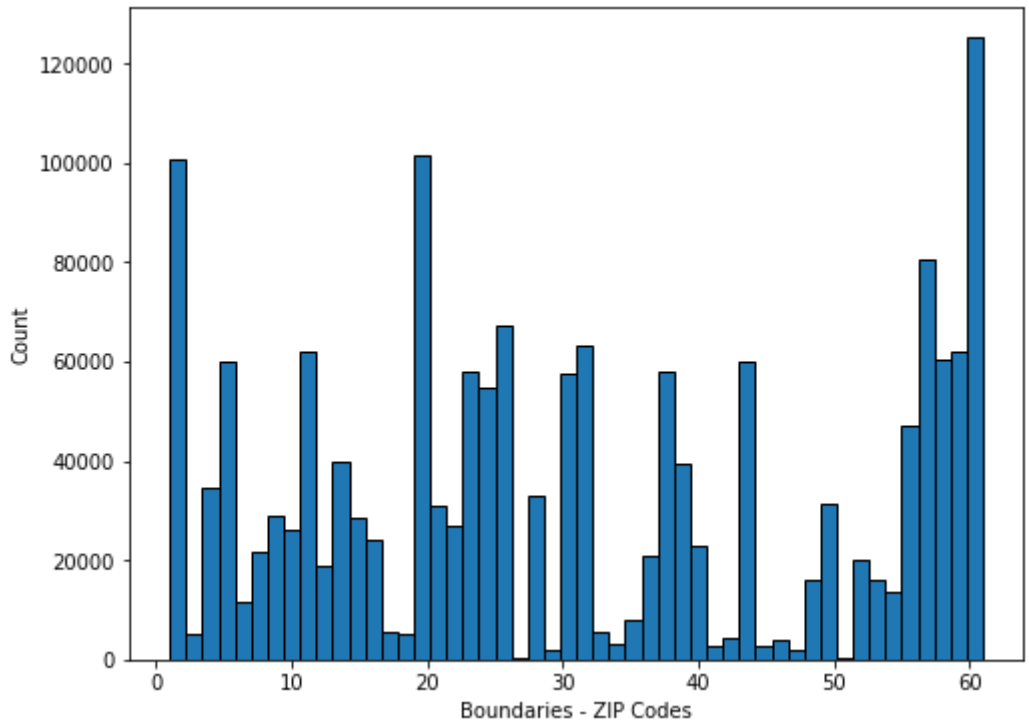
```
boxplot('Boundaries - ZIP Codes')
```



直方图

In [69]:

```
hist('Boundaries - ZIP Codes')
```



30.Census Tracts

In [70]:

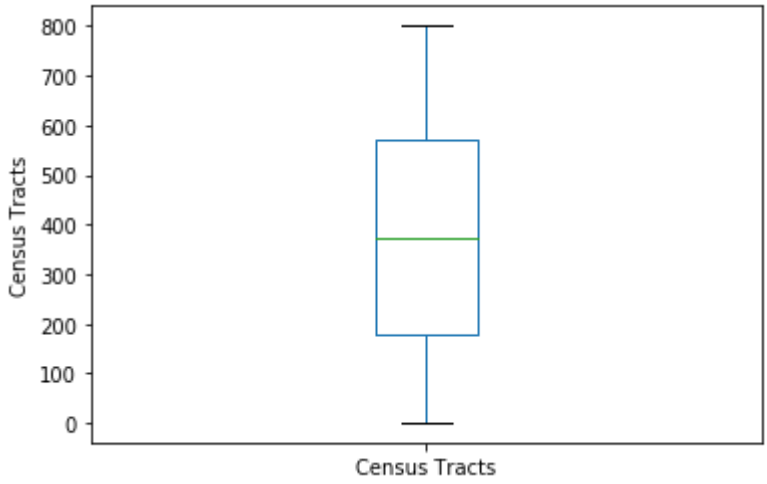
```
print('Census Tracts五数概括:',fiveNumber('Census Tracts'))
```

```
Census Tracts五数概括: [1.0, 179.0, 374.0, 572.0, 801.0, -410.5, 1161.5]
```

盒图

In [71]:

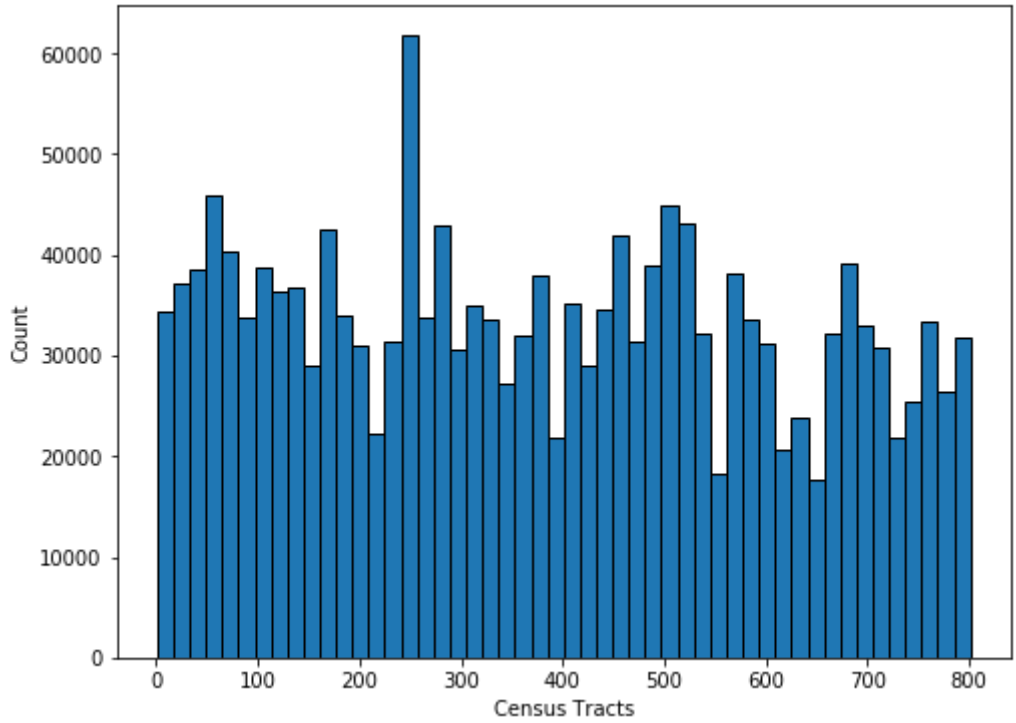
```
boxplot('Census Tracts')
```



直方图

In [72]:

```
hist('Census Tracts')
```



31.Wards

In [73]:

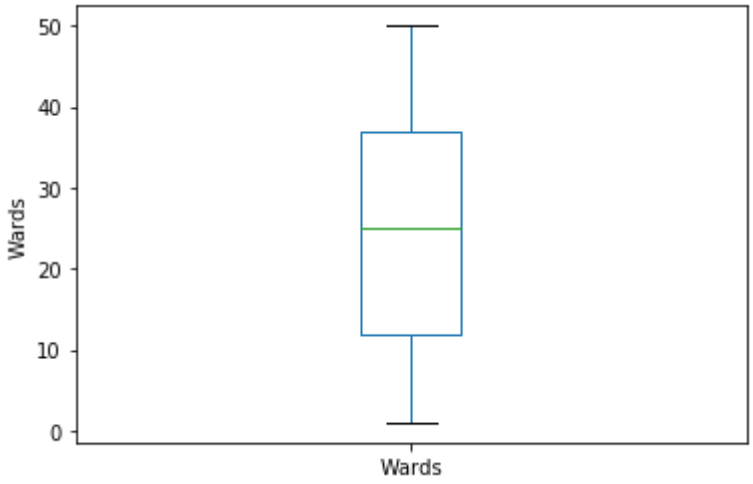
```
print('Wards五数概括:',fiveNumber('Wards'))
```

```
Wards五数概括: [1.0, 12.0, 25.0, 37.0, 50.0, -25.5, 74.5]
```

盒图

In [74]:

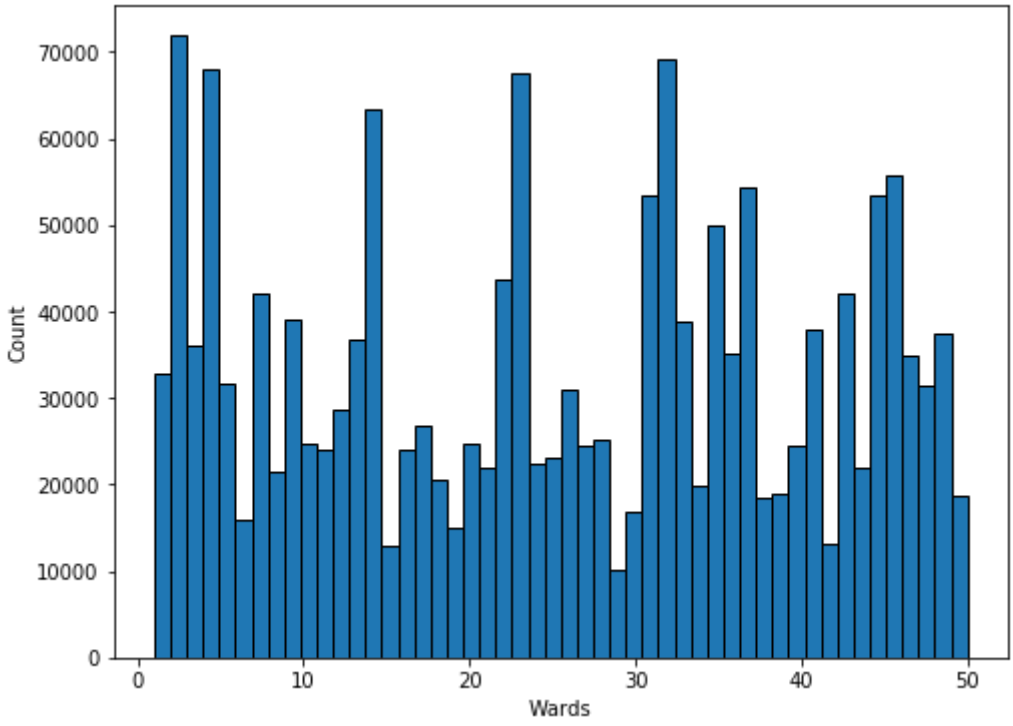
```
boxplot('Wards')
```



直方图

In [75]:

```
hist('Wards')
```



32.Historical Wards 2003-2015

In [76]:

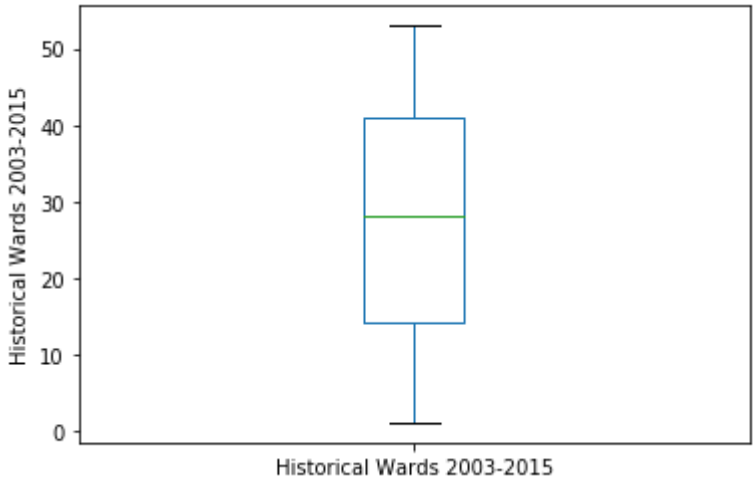
```
print('Historical Wards 2003-2015五数概括:',fiveNumber('Historical Wards 2003-2015'))
```

```
Historical Wards 2003-2015五数概括: [1.0, 14.0, 28.0, 41.0, 53.0, -26.5, 81.5]
```

盒图

In [77]:

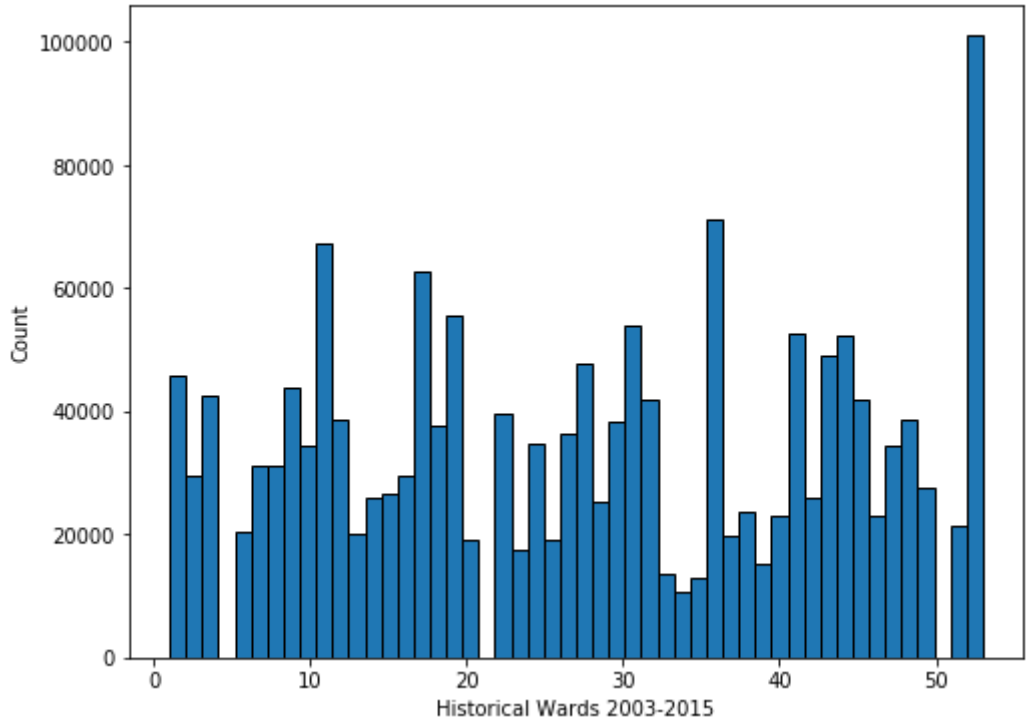
```
boxplot('Historical Wards 2003-2015')
```



直方图

In [78]:

```
hist('Historical Wards 2003-2015')
```



数据缺失处理

In [79]:

```
for i in data.columns.values.tolist():
    print(i,"缺失",data[i].isna().sum())
```

```
ID 缺失 0
VIOLATION LAST MODIFIED DATE 缺失 0
VIOLATION DATE 缺失 0
VIOLATION CODE 缺失 0
VIOLATION STATUS 缺失 0
VIOLATION STATUS DATE 缺失 1036199
VIOLATION DESCRIPTION 缺失 10768
VIOLATION LOCATION 缺失 897282
VIOLATION INSPECTOR COMMENTS 缺失 175463
VIOLATION ORDINANCE 缺失 47581
INSPECTOR ID 缺失 0
INSPECTION NUMBER 缺失 0
INSPECTION STATUS 缺失 16
INSPECTION WAIVED 缺失 0
INSPECTION CATEGORY 缺失 0
DEPARTMENT BUREAU 缺失 0
ADDRESS 缺失 0
STREET NUMBER 缺失 0
STREET DIRECTION 缺失 0
STREET NAME 缺失 0
STREET TYPE 缺失 13541
PROPERTY GROUP 缺失 0
SSA 缺失 1356267
LATITUDE 缺失 1510
LONGITUDE 缺失 1510
LOCATION 缺失 1510
Community Areas 缺失 2279
Zip Codes 缺失 1510
Boundaries - ZIP Codes 缺失 2279
Census Tracts 缺失 1545
Wards 缺失 2279
Historical Wards 2003-2015 缺失 2279
```

VIOLATION STATUS DATE 缺失 1036199
VIOLATION DESCRIPTION 缺失 10768
VIOLATION LOCATION 缺失 897282
VIOLATION INSPECTOR COMMENTS 缺失 175463
VIOLATION ORDINANCE 缺失 47581
INSPECTION STATUS 缺失 16
STREET TYPE 缺失 13541
SSA 缺失 1356267
LATITUDE 缺失 1510
LONGITUDE 缺失 1510
LOCATION 缺失 1510
Community Areas 缺失 2279
Zip Codes 缺失 1510
Boundaries - ZIP Codes 缺失 2279
Census Tracts 缺失 1545
Wards 缺失 2279
Historical Wards 2003-2015 缺失 2279

1.将缺失部分剔除

SSA缺失较多数据，且目前没有看出用处，故将其剔除

In [80]:

```
data.shape
```

Out[80]:

(1677788, 32)

In [81]:

```
del data['SSA']
data.shape
```

Out[81]:

(1677788, 31)

2.用最高频率值来填补缺失值

VIOLATION LOCATION缺失可以使用最有可能的值进行填充

In [83]:

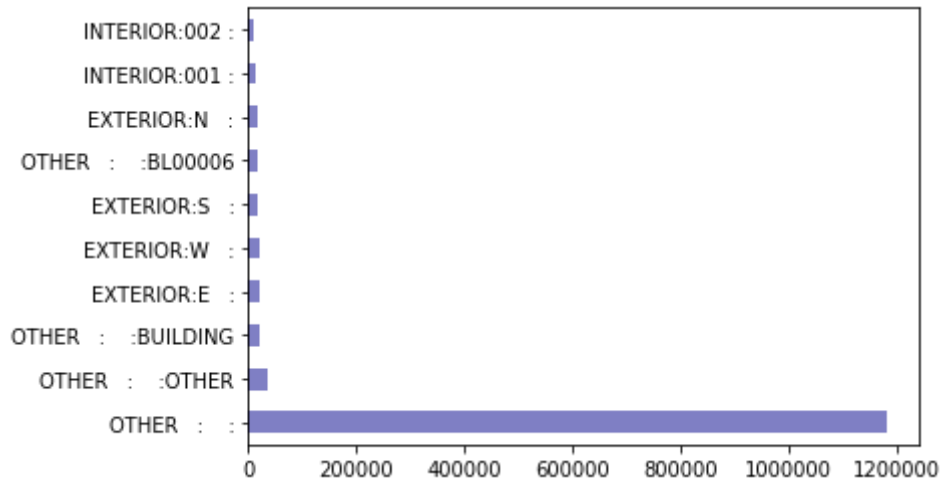
```
for index,row in data.iterrows():
    i=row['VIOLATION LOCATION']
    if i is np.nan:
        data.at[index,'VIOLATION LOCATION']=violocVC.index[0]
```

In [84]:

```
newviolocVC=data['VIOLATION LOCATION'].value_counts().head(10)
newviolocVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[84]:

<matplotlib.axes._subplots.AxesSubplot at 0x26f525bd1c8>



3.通过属性的相关关系来填补缺失值

streetType可以通过在address中获取，如果address缺失该数据，则用unkown填充

In [85]:

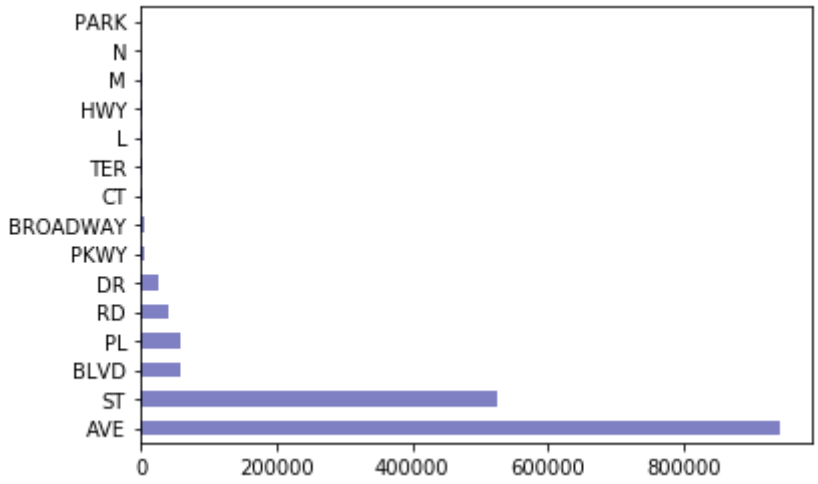
```
for index,row in data.iterrows():
    j=row['STREET TYPE']
    if j is np.nan:
        k=row['ADDRESS']
        data.at[index,'STREET TYPE']=k.split()[-1]
```

In [86]:

```
newstVC=data['STREET TYPE'].value_counts().head(15)
newstVC.plot(kind='barh', color='darkblue', alpha=0.5)
```

Out[86]:

<matplotlib.axes._subplots.AxesSubplot at 0x27024b8d048>



In [87]:

```
newstVC
```

Out[87]:

```
AVE          940725
ST           523743
BLVD          59536
PL            57665
RD            41100
DR            27145
PKWY           6605
BROADWAY       5210
CT             3287
TER            2222
L              1926
HWY            1559
M              1492
N               801
PARK            793
Name: STREET TYPE, dtype: int64
```

4.通过数据对象之间的相似性来填补缺失值

按照LONGITUDE和LATITUDE进行分组和排序，当location缺失时可以直接用相邻的location进行填充

In [ ]:

```
newData=data.groupby('LONGITUDE',sort=False).apply(lambda x:x.sort_values('LATITUDE',ascending
=True)).reset_index(drop=True)
```

In [ ]:

```

```