互评作业1：数据探索性分析与数据预处理：Chicago Building Violations

学号：1120183560 姓名：刘文楷

1. 数据集：wine-reviews

1个csv文件:

- building-violations.csv

读取数据:

In [18]:

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

In [19]:

```
dirpath_cbv = "ChicagoBuildingViolations-dat
```

In [21]:

```
data_cbv = pd.read_csv(dirpath_cbv)
data_cbv.head()
```

	ID	VIOLATION LAST MODIFIED DATE	VIOLATION DATE
0	6392482	2019-12-04T12:40:09.000	2019-12-04T00:00:00.000
1	6392480	2019-12-04T12:40:09.000	2019-12-04T00:00:00.000
2	6392335	2019-12-04T14:00:12.000	2019-12-04T00:00:00.000
3	6391883	2019-12-04T08:32:01.000	2019-12-04T00:00:00.000
4	6392369	2019-12-04T14:14:24.000	2019-12-04T00:00:00.000

```
5 rows × 32 columns
```

显示数据信息

In [22]:

```
data_cbv = data_cbv.drop(columns=['ID', 'VIO


data_cbv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1677788 entries, 0 to 1677
787
Data columns (total 29 columns):
 #   Column                        Non
-Null Count    Dtype
---  ------                        ---
-----------    -----
 0   VIOLATION LAST MODIFIED DATE  167
7788 non-null  object
 1   VIOLATION DATE                167
7788 non-null  object
 2   VIOLATION CODE                167
7788 non-null  object
 3   VIOLATION STATUS              167
7788 non-null  object
 4   VIOLATION STATUS DATE         641
589 non-null   object
 5   VIOLATION LOCATION            780
506 non-null   object
 6   VIOLATION ORDINANCE           163
0207 non-null  object
 7   INSPECTOR ID                  167
7788 non-null  object
 8   INSPECTION NUMBER             167
7788 non-null  int64
 9   INSPECTION STATUS             167
7772 non-null  object
 10  INSPECTION WAIVED             167
7788 non-null  object
 11  INSPECTION CATEGORY           167
7788 non-null  object
 12  DEPARTMENT BUREAU             167
7788 non-null  object
 13  ADDRESS                       167
7788 non-null  object
```

```
 14  STREET NUMBER               167
7788 non-null  int64
 15  STREET DIRECTION            167
7788 non-null  object
 16  STREET NAME                 167
7788 non-null  object
 17  STREET TYPE                 166
4247 non-null  object
 18  PROPERTY GROUP              167
7788 non-null  int64
 19  SSA                         321
521 non-null   float64
 20  LATITUDE                    167
6278 non-null  float64
 21  LONGITUDE                   167
6278 non-null  float64
 22  LOCATION                    167
6278 non-null  object
 23  Community Areas             167
5509 non-null  float64
 24  Zip Codes                   167
6278 non-null  float64
 25  Boundaries - ZIP Codes      167
5509 non-null  float64
 26  Census Tracts               167
6243 non-null  float64
 27  Wards                       167
5509 non-null  float64
 28  Historical Wards 2003-2015    167
5509 non-null  float64
dtypes: float64(9), int64(3), object(1
7)
memory usage: 371.2+ MB
```

2. 数据分析

2.1 数据可视化和摘要

2.1.1 标称属性

In [23]:

```
title = ['VIOLATION LAST MODIFIED DATE', 'VI
```

In [24]:

```
# 由于本数据集属性太多，这里仅显示violation statu
title[3]
```

```
'VIOLATION STATUS'
```

In [25]:

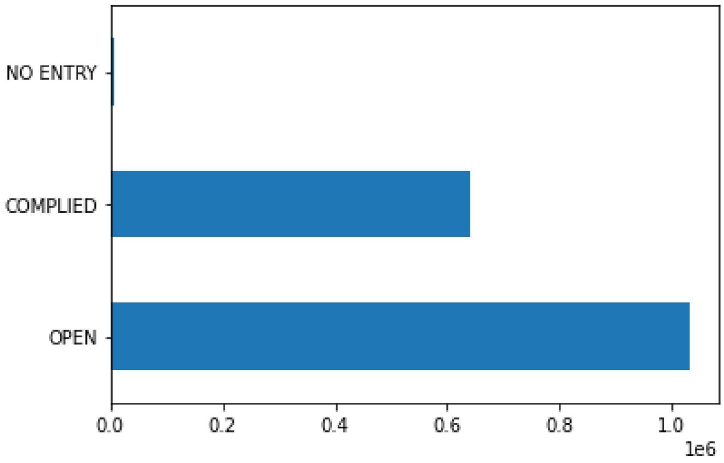
```
getattr(data_cbv, title[3]).value_counts()
```

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

In [26]:

```
data_cbv[title[3]].value_counts().head(10).p
```

```
<AxesSubplot:>
```



2.2.2 数值属性

直接调用describe函数给出数据的基本统计量

In [27]:

```
data_cbv.describe()
```

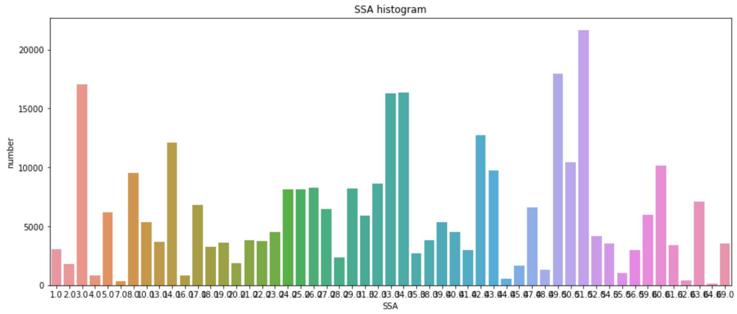
	INSPECTION NUMBER	STREET NUMBER	PROPER GRO
count	1.677788e+06	1.677788e+06	1.677788e+
mean	8.049798e+06	4.150382e+03	2.020547e+
std	4.555757e+06	2.893493e+03	1.862796e+
min	2.655750e+05	1.000000e+00	1.000000e+
25%	2.304416e+06	1.648000e+03	2.056000e+
50%	1.041875e+07	3.747000e+03	1.543230e+
75%	1.168728e+07	6.228000e+03	3.669840e+
max	1.305092e+07	1.377000e+04	6.779750e+

数据可视化

直方图:

In [28]:

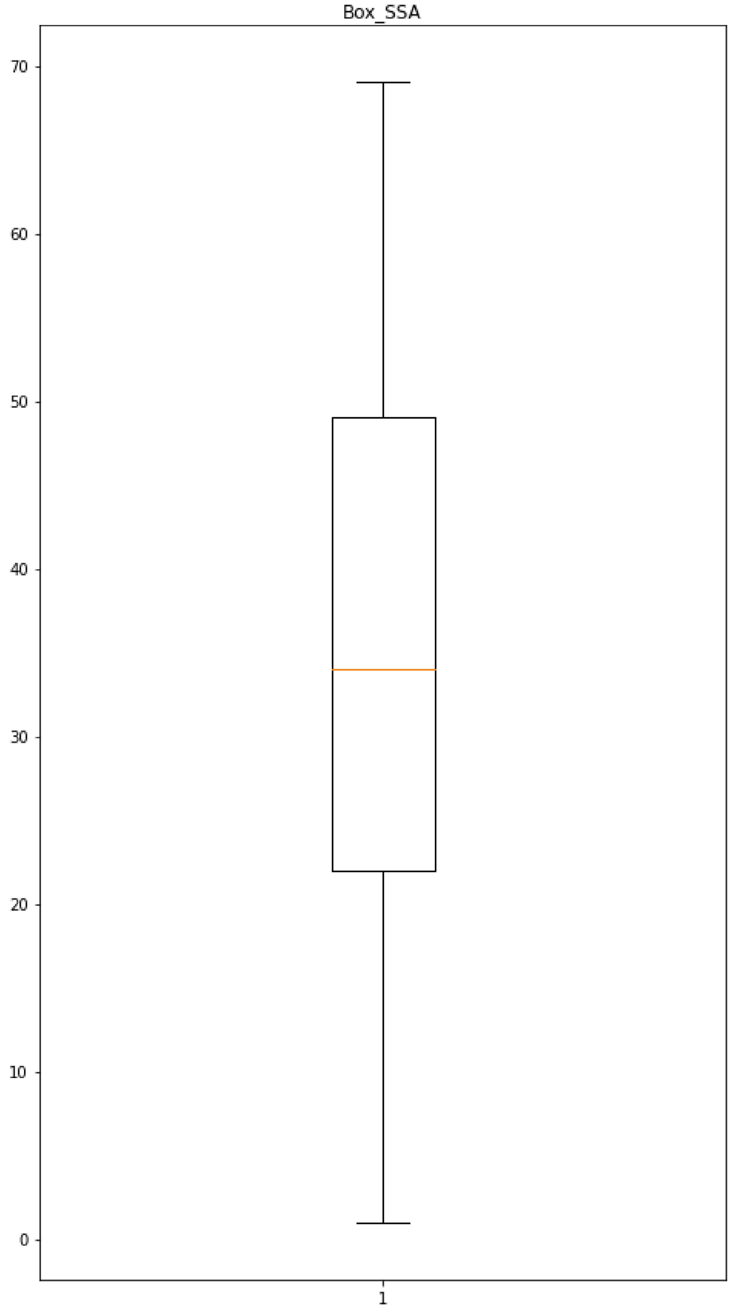
```
def histogram(data, x, y ,title):
    plt.figure(figsize = (15,6))
    plt.title(title)
    sns.set_color_codes("pastel")
    sns.barplot(x=x, y=y, data=df)
    locs, labels = plt.xticks()
    plt.show()

#SSA
temp = data_cbv['SSA'].value_counts()
df = pd.DataFrame({'SSA':temp.index, 'number

histogram(df, 'SSA', 'number', 'SSA histogra
```



盒图:

In [30]:

```
fig = plt.figure(figsize=(8, 15))
plt.boxplot(data_cbv['SSA'].loc[data_cbv['SS
t = plt.title('Box_SSA')
```

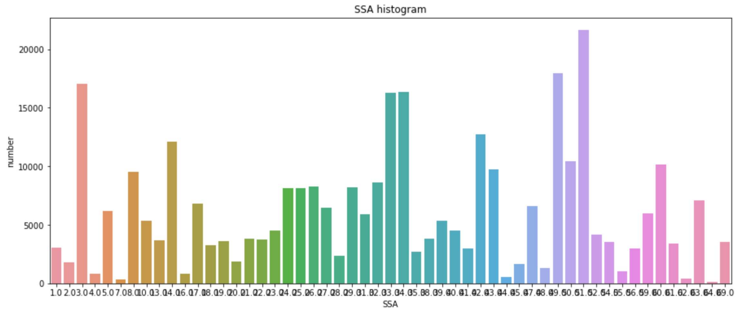


3. 数据缺失处理

3.1 将缺失部分剔除

In [31]:

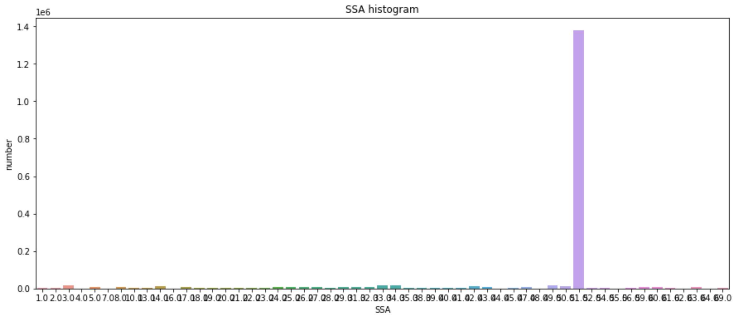
```
data_delete = data_cbv['SSA'].dropna()

temp = data_delete.value_counts()
df = pd.DataFrame({'SSA':temp.index, 'number

histogram(df, 'SSA', 'number', 'SSA histogra
```



3.2 用最高频率来填补缺失值

In [34]:

```
data_most = data_cbv['SSA'].fillna(data_cbv[
temp = data_most.value_counts()
df = pd.DataFrame({'SSA':temp.index, 'number
histogram(df, 'SSA', 'number', 'SSA histogra
```



3.3 通过属性的相关关系来填补缺失值

使用纬度LATITUDE填补经度LONGITUDE

In [35]:

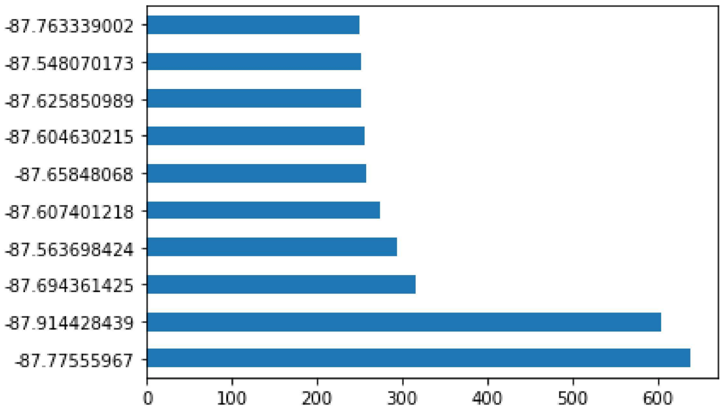
```
data_fill = pd.DataFrame(data_cbv, columns=[
data_fill.head(10)
```

	LATITUDE	LONGITUDE
0	41.749169	-87.602551
1	41.749169	-87.602551
2	41.711751	-87.537842
3	41.844521	-87.712416
4	41.753908	-87.562784
5	41.806815	-87.611539
6	41.753908	-87.562784
7	41.749169	-87.602551
8	41.711751	-87.537842
9	41.748732	-87.659904

In [36]:

```
data_fill['LONGITUDE'].value_counts().head(1
```

```
<AxesSubplot:>
```



3.4 通过数据对象之间的相似性来填补缺失值

In [38]:

```
data_sim = data_cbv[['LATITUDE','LONGITUDE']
point2price = {}
for row in data_sim.iterrows():
    if point2price.get(row[1]['LONGITUDE'],
        if not pd.isnull(row[1]['LATITUDE'])
            point2price[row[1]['LONGITUDE']]
            point2price[row[1]['LONGITUDE']]
    else:
        if not pd.isnull(row[1]['LATITUDE'])
            point2price[row[1]['LONGITUDE']]
```

In [39]:

```
for k in point2price.keys():
    point2price[k][0] = round(point2price[k]
```

In [40]:

```
for row in data_sim.iterrows():
    if pd.isnull(row[1]['LATITUDE']):
        try:
            row[1]['LATITUDE'] = point2price
        except:
            continue
```

In [*]:

```
#对被填充后的price画直方图
temp = data_sim['LATITUDE'].value_counts()
df = pd.DataFrame({'LATITUDE':temp.index, 'n

histogram(df, 'LATITUDE', 'number', 'LATITUD
```

In [ ]: