

1. 数据脱敏，将违章者的姓名、驾驶证号等个人隐私信息替换；
2. 交通违章行为危害性评估：根据违章的类型、描述、事故等级等信息，给每次违章设置一个危害指数；
3. 提取违章者的违法行为特征变量I1,I2,I3...如司机在一年内违章的次数、两次违章之间间隔、违章危害指数等；
4. 对违章者数据进行聚类：对3中的特征变量进行归一化、划分等级、以及相关性评估，选取用于聚类的变量，利用k-means进行聚类。得到k个违章者群体。
5. 分别对k个违章群体进行描述性分析：属于这个群体的违章者的主要准驾车辆、违章时间分布、主要违章行为等信息，词云可视化；
6. 选取白天/夜间，工作日/周末等研究区间，进行4-5步骤，探究在不同的时间段内违章群体的构成与特征是否会发生变化。

```

14 import pandas as pd
from IPython.display import display
import numpy as np

from matplotlib import pyplot as plt

15 # data_path = "/exstorage/zjr/traffic/data/data3.csv"
data_path = "/Users/jiarui/Study/交通事故/data/data3.csv"
df = pd.read_csv(data_path)
df.columns

/opt/anaconda3/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3146: DtypeWarning:
has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
15 Index(['唯一标识', '违法来源', '违法编号', '违法时间', '行政区划', '当事人_姓名', '当事人_驾驶证号',
       '当事人_档案编号', '当事人_发证机关', '当事人_准驾车型', '机动车_号牌种类', '机动车_号牌种类说明',
       '机动车_号牌号码', '机动车_所有人', '机动车_使用性质', '机动车_使用性质说明', '机动车_交通方式',
       '机动车_交通方式说明', '违法地点', '违法地点_道路类型', '违法地点_路段代码', '违法地点_地点米数',
       '违法地点_违法地址', '违法地点_公路行政等级', '违法行为', '违法行为_违法大类', '违法行为_违法小类',
       '违法行为_违法描述', '违法行为_事故等级', '违法行为记分数', '违法行为罚款金额', '违法行为_实测值',
       '违法行为_缴款方式', '违法行为_缴款标记', '违法行为_缴款日期', '违法执勤_发现机关部门代码', '违法执
       '违法执勤_发现机关支队名称', '违法执勤_发现机关大队名称', '违法执勤_发现机关中队名称', '违法执勤_处理
       '违法执勤_处理机关部门名称', '违法执勤_处理机关支队名称', '违法执勤_处理机关大队名称', '违法执勤_处理
       '违法执勤_处理时间', '违法执勤_执勤民警姓名', '违法执勤_执勤民警警号', '违法执勤_录入人', '违法执勤_
       '违法处理_是否强措', '违法处理_裁决标记', '违法处理_裁决时间', '违法处理_是否吊销', '违法处理_吊销原
       '违法处理_吊销原因描述', '违法处理_吊销日期', '违法处理_是否拘留', '违法处理_拘留期限', '违法处理_扣
       '违法处理_拘留终止日期', '违法处理_是否暂扣', '违法处理_暂扣原因', '违法处理_暂扣期限', '违法处理_扣
       '违法处理_扣证结束日期', '记录类型', '凭证编号', '决定书编号', '决定书类别', '录入时间', '更新时间
       dtype='object')

```

数据预处理

数据脱敏

选择有需要的列

筛选出机动车和非机动车，机动车可以根据驾驶证号计算年龄

```
16 # #选取前10000条数据 给老师展示
```

```

# import uuid

# df = df.head(10000)
# list_ids = list(df['当事人_驾驶证号'].unique())
# for uid in list_ids:
#     df.loc[df['当事人_驾驶证号'] == uid, '当事人_驾驶证号'] = uuid.uuid1()
# list_ids = list(df['当事人_驾驶证号'].unique())

17 # df = df.drop('当事人_姓名', axis=1)
# df.head()

18 # df.to_csv("/exstorage/zjr/traffic/data/data_preprocess_gbk.csv", encoding = 'gbk')

19 ## 选择有需要的列
df = df[['唯一标识', '违法时间', '行政区划', '当事人_姓名', '当事人_驾驶证号',
         '当事人_发证机关', '当事人_准驾车型', '机动车_号牌种类', '机动车_号牌种类说明', '机动车_使用性质',
         '机动车_交通方式说明', '违法地点', '违法地点_道路类型', '违法地点_路段代码', '违法地点_地点米数',
         '违法地点_违法地址', '违法地点_公路行政等级', '违法行为', '违法行为_违法大类', '违法行为_违法小类',
         '违法行为_违法描述', '违法行为_事故等级', '违法行为记分数', '违法行为罚款金额', '违法行为_实测值']

20 list_display = ['当事人_驾驶证号', '违法时间', '机动车_交通方式说明', '违法行为_违法小类', '违法行为_违法

21 ## 筛选机动车与非机动车
pd.set_option('display.max_rows', 200)
action_count = df['机动车_交通方式说明'].value_counts()
# print(action_count.head(104))
print(len(action_count))

# action_count = df['机动车_号牌种类说明'].value_counts()
# print(action_count.head(104))
# print(len(action_count))
list_no_vehicle = ['步行', '乘车', '电动自行车', '其他非机动车', '自行车', '三轮车']
df_no_vehicle = df[df['机动车_交通方式说明'].isin(list_no_vehicle)]
df_vehicle = df.drop(df[df['机动车_交通方式说明'].isin(list_no_vehicle)].index)

action_count = df_no_vehicle['机动车_交通方式说明'].value_counts()
print(action_count.head(104))
print(len(action_count))

action_count = df_vehicle['机动车_交通方式说明'].value_counts()
print(action_count.head(104))
print(len(action_count))

action_count = df_vehicle['机动车_交通方式'].value_counts()
print(action_count.head(104))
print(len(action_count))

104
电动自行车      48465
步行          20199
其他非机动车    18093
乘车          14685
三轮车          2982
自行车          104
Name: 机动车_交通方式说明, dtype: int64
6
小型轿车      84531
轻型栏板货车    81453
重型自卸货车    65606

```

重型半挂牵引车	62542
重型厢式货车	11617
中型栏板货车	10932
重型非载货专项作业车	10397
轻型厢式货车	9299
重型仓栅式货车	6346
轻型自卸货车	6237
小型面包车	4719
轻便二轮摩托车	4307
自卸低速货车	4008
小型普通客车	3540
中型自卸货车	3468
中型普通客车	2845
大型轮式拖拉机	2510
小型越野客车	2399
中型厢式货车	2280
重型罐式货车	2146
小型轮式拖拉机	1905
正三轮载货摩托车	1576
普通二轮摩托车	1390
重型自卸半挂车	895
手扶拖拉机	724
轻型仓栅式货车	640
重型栏板半挂车	589
轻型封闭式货车	577
正三轮载客摩托车	551
轻便正三轮摩托车	448
重型仓栅式半挂车	436
微型栏板货车	430
微型面包车	370
微型轿车	368
三轮汽车	351
微型普通客车	346
普通正三轮摩托车	285
重型栏板货车	261
重型平板货车	250
其它	185
栏板低速货车	147
中型罐式货车	143
小型非载货专项作业车	137
大型卧铺客车	135
中型半挂牵引车	104
手扶变形运输机	100
中型栏板半挂车	89
轻型栏板半挂车	85
大型普通客车	81
中型非载货专项作业车	73
微型厢式货车	70
重型集装箱半挂车	63
重型罐式半挂车	42
重型专项作业半挂车	41
重型厢式全挂车	40
重型平板半挂车	40
重型全挂牵引车	36
残疾人专用车	31
重型封闭式货车	29
重型特殊结构货车	23
重型厢式半挂车	21
中型自卸半挂车	19
轻型特殊结构货车	15
重型低平板半挂车	13
中型自卸全挂车	12
微型自卸货车	12
大型非载货专项作业车	11
大型双层客车	9
轻型栏板全挂车	8
中型平板货车	8
重型栏板全挂车	8

中型专用客车	6
轻型自卸半挂车	6
厢式低速货车	5
重型自卸全挂车	5
助力自行车	5
轻型平板货车	4
轻型低平板半挂车	3
中型特殊结构货车	3
手推车	3
微型封闭式货车	2
中型集装箱半挂车	2
轻型厢式全挂车	2
大型专用校车	2
轮式挖掘机械	2
微型越野客车	2
轻型厢式半挂车	2
中型低平板半挂车	2
中型厢式全挂车	2
中型仓栅式货车	2
有轨电车	1
小型专用客车	1
中型罐式半挂车	1
大型轿车	1
中型厢式半挂车	1
大型越野客车	1
中型集装箱全挂车	1
微型非载货专项作业车	1

Name: 机动车_交通方式说明, dtype: int64

98

K33	84531
H31	81453
H17	65606
Q11	62542
H12	11617
H21	10932
Z51	10397
H32	9299
H19	6346
H37	6237
K39	4719
M22	4307
H54	4008
K31	3540
H27	3468
K21	2845
T11	2510
K32	2399
H22	2280
H14	2146
T21	1905
M14	1576
M21	1390
B16	895
T22	724
H39	640
B11	589
H33	577
M13	551
M12	448
B18	436
H41	430
K49	370
K43	368
N11	351
K41	346
M11	285
H11	261
H15	250

```
X99      185
H51      147
H24      143
Z31      137
K13      135
Q21      104
T23      100
B21      89
B31      85
K11      81
Z21      73
H42      70
B15      63
B13      42
B1A      41
G12      40
B14      40
Q12      36
F04      31
H13      29
H18      23
B12      21
B26      19
H38      15
B1B      13
H45      12
G26      12
Z11      11
K12      9
G31      8
H25      8
G11      8
K27      6
B35      6
F06      5
G16      5
H52      5
H35      4
H28      3
B39      3
F03      3
B25      2
K18      2
K42      2
B32      2
J12      2
G22      2
G32      2
H43      2
B2B      2
H29      2
K16      1
Z41      1
B23      1
B22      1
K15      1
G25      1
K34      1
D12      1
Name: 机动车_交通方式, dtype: int64
```

98

```
22 # 区分营运车辆和非营运车辆
df_vehicle['机动车_使用性质说明'].value_counts()

22 非营运      218535
    货运        165130
    营转非      4039
```

出租客运	3582
危化品运输	1203
公路客运	847
教练	258
公交客运	79
出租转非	59
租赁	59
旅游客运	58
工程救险	55
救护	4
警用	3
消防	3
小学生校车	1

Name: 机动车_使用性质说明, dtype: int64

```

23 # print(df_vehicle[df_vehicle['机动车_使用性质说明'] == '非营运']['机动车_交通方式说明'].value_count)

# print(df_vehicle[df_vehicle['机动车_使用性质说明'] == '租赁']['机动车_交通方式说明'].value_counts())
df_no_yingyun = df_vehicle[df_vehicle['机动车_使用性质说明'] == '非营运']
df_yingyun = df_vehicle[df_vehicle['机动车_使用性质说明'] != '非营运']

24 # 计算dataframe的各个维度特征, 如交通方式编码、计算年龄、给违法行为评估
def data_get_feature(df_vehicle):
    ## 直接去掉没有驾驶证号的
    print('原来机动车违法的数据有{}条'.format(len(df_vehicle)))
    df_vehicle = df_vehicle.drop(df_vehicle[df_vehicle['当事人_驾驶证号'].isin(['无', np.nan])].index)
    # 删除掉驾驶证号不完全或者有误的
    df_vehicle = df_vehicle.loc[df_vehicle['当事人_驾驶证号'].str.len() == 18].reset_index()
    print('经过去除无驾驶证号之后, 还有{}'.format(len(df_vehicle)))
    df_vehicle[['当事人_姓名', '当事人_驾驶证号',
                '当事人_发证机关', '当事人_准驾车型', '机动车_交通方式说明', '违法地点']]

    # 给交通方式编码
    df_vehicle['trans_code'] = 0
    # print(df_vehicle['机动车_交通方式说明'].str.contains('小型轿车'))
    list_1 = ['小型轿车', '面包车']
    df_vehicle.loc[df_vehicle['机动车_交通方式说明'].str.contains('小型轿车|面包车'), 'trans_code'] = 1

    list_2 = ['货车', '大型轿车', '作业车', '大型专用校车']
    df_vehicle.loc[df_vehicle['机动车_交通方式说明'].str.contains('货车|大型轿车|作业车|大型专用校车|'), 'trans_code'] = 2

    list_3 = ['牵引车', '挂车']
    df_vehicle.loc[df_vehicle['机动车_交通方式说明'].str.contains('牵引车|挂车|轮式挖掘机械'), 'trans_code'] = 3

    list_4 = ['客车']
    df_vehicle.loc[df_vehicle['机动车_交通方式说明'].str.contains('客车'), 'trans_code'] = 4

    list_5 = ['摩托车']
    df_vehicle.loc[df_vehicle['机动车_交通方式说明'].str.contains('摩托车'), 'trans_code'] = 5

    df_vehicle.loc[df_vehicle['机动车_交通方式说明'].str.contains('微型轿车|三轮汽车|残疾人专用车|助步器'), 'trans_code'] = 6

    print("-----交通方式编码完成-----")

    ## 计算年龄
    born_year = df_vehicle['当事人_驾驶证号'].str[6:10]
    # born_year = born_year.astype('int64', errors = 'ignore')
    # pd.to_numeric(born_year, errors='coerce').fillna(0)
    df_vehicle['age'] = 2013 - pd.to_numeric(born_year, errors='coerce').fillna(0)
    df_vehicle = df_vehicle.drop(
        df_vehicle[(df_vehicle['age'] > 80) | (df_vehicle['age'] < 10) | (df_vehicle['age'] == 0)].index)

    # 筛选违法行为类型
    df_vehicle = df_vehicle.drop(
        df_vehicle[(df_vehicle['违法行为_违法小类'] == '其他影响安全行为') | (df_vehicle['违法行为_违法大类'] == '违反规定行驶')].index)

```

```

df_vehicle = df_vehicle.drop(
    df_vehicle[(df_vehicle['违法行为_违法小类'] == '其他') | (df_vehicle['违法行为_违法小类'] == '驾驶人未按规定使用安全带')]
)
df_vehicle = df_vehicle.drop(
    df_vehicle[(df_vehicle['违法行为_违法小类'] == '铁路道口或渡口') | (df_vehicle['违法行为_违法小类'] == '行人、非机动车、牲畜冲撞机动车')]
)
df_vehicle = df_vehicle.drop(
    df_vehicle[(df_vehicle['违法行为_违法小类'] == '违法牵引') | (df_vehicle['违法行为_违法小类'] == '违反交通信号') & (df_vehicle['违法行为_违法描述'].str.contains('违反道路交通信号灯'))], '违法行为_违法小类' = ''
)

print(len(df_vehicle))
#     print(df_vehicle['违法行为_违法小类'].value_counts())

#给违法行为的危险性打分
df_vehicle['occur_prob'] = 1.0
df_vehicle['severity'] = 1.0

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法上道路行驶', 'occur_prob'] = 1
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法上道路行驶', 'severity'] = 1

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法停车', 'occur_prob'] = 1
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法停车', 'severity'] = 1

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违反交通信号', 'occur_prob'] = 1
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违反交通信号', 'severity'] = 1
df_vehicle.loc[
    (df_vehicle['违法行为_违法小类'] == '违反交通信号') & (df_vehicle['违法行为_违法描述'].str.contains('行人、非机动车、牲畜冲撞机动车'))], '违法行为_违法小类' = ''
df_vehicle.loc[
    (df_vehicle['违法行为_违法小类'] == '违反交通信号') & (df_vehicle['违法行为_违法描述'].str.contains('违反道路交通信号灯'))], '违法行为_违法小类' = ''
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违反交通信号灯', 'severity'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违反交通信号灯', 'occur_prob'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法装载', 'occur_prob'] = 2
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法装载', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '未按规定办理业务', 'occur_prob'] = 1
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '未按规定办理业务', 'severity'] = 1

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法占道行驶', 'occur_prob'] = 3
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法占道行驶', 'severity'] = 3

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法抢行', 'occur_prob'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法抢行', 'severity'] = 4

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '货动车辆超载', 'occur_prob'] = 2
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '货动车辆超载', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法装载', 'occur_prob'] = 2
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法装载', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '不按规定使用灯光', 'occur_prob'] = 2
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '不按规定使用灯光', 'severity'] = 3

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '饮酒驾驶', 'occur_prob'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '饮酒驾驶', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '超速行驶', 'occur_prob'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '超速行驶', 'severity'] = 4

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法掉头', 'occur_prob'] = 4
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法掉头', 'severity'] = 4

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '疲劳驾驶', 'occur_prob'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '疲劳驾驶', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '无证驾驶', 'occur_prob'] = 4
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '无证驾驶', 'severity'] = 3

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '逆行', 'occur_prob'] = 5

```

```

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '逆行', 'severity'] = 4

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '未按规定让行', 'occur_prob'] = 1
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '未按规定让行', 'severity'] = 1

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法倒车', 'occur_prob'] = 2
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法倒车', 'severity'] = 2

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '醉酒驾驶', 'occur_prob'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '醉酒驾驶', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法变更车道', 'occur_prob'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法变更车道', 'severity'] = 4

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '客运车辆超员', 'occur_prob'] = 2
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '客运车辆超员', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '未低速通过', 'occur_prob'] = 4
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '未低速通过', 'severity'] = 4

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法超车', 'occur_prob'] = 4
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法超车', 'severity'] = 3

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法装载超限及危险品运输', 'occur_prob'] = 2
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法装载超限及危险品运输', 'severity'] = 5

df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法会车', 'occur_prob'] = 5
df_vehicle.loc[df_vehicle['违法行为_违法小类'] == '违法会车', 'severity'] = 4

# 外来或本地,省内为1 省外为2
df_vehicle['is_local'] = 1
df_vehicle.loc[df_vehicle['机动车_号牌号码'].str[0:1] != '浙', 'is_local'] = 2
df_vehicle['is_local'].value_counts()

#数据概况
driver_count = df_vehicle['当事人_驾驶证号'].value_counts()
count = (driver_count >= 3).sum()
count2 = (driver_count == 2).sum()
print(
    "机动车总共违法记录{}, 总共有{}个违法人\n其中违法次数大于等于3的有{}人\n违法次数2的有{}人".format(
        count,
        count2,
        count - count2,
        count2
    )
)
return df_vehicle

```

25 df_no_yingyun_value = data_get_feature(df_no_yingyun)
df_yingyun_value = data_get_feature(df_yingyun)

原来机动车违法的数据有218535条

经过去除无驾驶证号的之后，还有217350

-----交通方式编码完成-----

158009

机动车总共违法记录158009，总共有119750个违法人

其中违法次数大于等于3的有7104人

违法次数2的有16075人

原来机动车违法的数据有176937条

经过去除无驾驶证号的之后，还有176878

-----交通方式编码完成-----

108893

机动车总共违法记录108893，总共有51771个违法人

其中违法次数大于等于3的有9311人

违法次数2的有9036人

26 df_yingyun_value.to_csv("/Users/jiarui/Study/交通事故/data/output/df_yingyun_value.csv")
df_no_yingyun_value.to_csv("/Users/jiarui/Study/交通事故/data/output/df_no_yingyun_value.csv")

选取聚类的特征变量

基本属性：本地/外来，驾驶车型、年龄；

行为属性：个人在一年内的违章次数、罚款金额、总扣分数、违章危害得分等；

```
27 def get_level(df_vehicle):
    df_calculate = df_vehicle[
        ['当事人_驾驶证号', 'age', 'is_local', 'trans_code', 'occur_prob', 'severity', '违法行为id']
    ]
    df_res = df_calculate.groupby('当事人_驾驶证号').agg('mean')
    df_res.columns = ['age', 'is_local', 'trans_code', 'occur_prob', 'severity', 'score', 'fine']
    df_res.index_name = '当事人_驾驶证号'
    df_res['counts'] = df_calculate['当事人_驾驶证号'].value_counts()
    # df_res[df_res['counts'] < 20]['counts'].plot(kind='hist')
    # 如果只是df_level=df_res 是浅拷贝，和引用类似，会导致修改原来的值
    df_level = df_res.copy(deep=True)
    df_level = df_level.round(0)
    list_col = df_res.columns
    # list_col = ['counts']
    for col in list_col:
        if col == 'id' or col == 'trans_code' or col == 'is_local':
            continue
        dfi = df_level[col].copy(deep=True)
        Q1 = df_level[col].quantile(0.25)
        Q2 = df_level[col].quantile(0.5)
        Q3 = df_level[col].quantile(0.75)
        IQR = Q3 - Q1
        Max = Q3 + IQR * 1.5
        Min = Q1 - IQR * 1.5
        if col == 'counts':
            df_level.loc[dfi >= 5, col] = 5
        elif col == 'score':
            df_level.loc[dfi <= 0.5, col] = 1
            df_level.loc[(dfi < 1) & (dfi > 0.5), col] = 2
            df_level.loc[(dfi < 2) & (dfi >= 1), col] = 3
            df_level.loc[(dfi < 3) & (dfi >= 2), col] = 4
            df_level.loc[(dfi < 6) & (dfi >= 3), col] = 5
            df_level.loc[dfi >= 6, col] = 6
        elif col == 'fine':
            df_level.loc[dfi < 50, col] = 1
            df_level.loc[(dfi < 100) & (dfi >= 50), col] = 2
            df_level.loc[(dfi < 150) & (dfi >= 100), col] = 3
            df_level.loc[(dfi < 200) & (dfi >= 150), col] = 4
            df_level.loc[(dfi < 300) & (dfi >= 200), col] = 5
            df_level.loc[dfi >= 300, col] = 6
        elif col == 'age':
            df_level.loc[dfi <= 25, col] = 1
            df_level.loc[(dfi < 35) & (dfi >= 25), col] = 2
            df_level.loc[(dfi < 45) & (dfi >= 35), col] = 3
            df_level.loc[(dfi < 55) & (dfi >= 45), col] = 4
            df_level.loc[dfi >= 55, col] = 5
        elif col == 'accident':
            df_level.loc[dfi == 0, col] = 1
            df_level.loc[(dfi <= 2) & (dfi > 0), col] = 2
            df_level.loc[(dfi <= 4) & (dfi > 2), col] = 3
        # else:
        #     df_level.loc[(dfi < Min) & (dfi >= 0), col] = 1
        #     df_level.loc[(dfi < Q1) & (dfi >= Min), col] = 2
        #     df_level.loc[(dfi < Q2) & (dfi >= Q1), col] = 3
        #     df_level.loc[(dfi < Q3) & (dfi >= Q2), col] = 4
        #     df_level.loc[(dfi < Max) & (dfi >= Q3), col] = 5
        #     df_level.loc[dfi >= Max, col] = 6
    df_level.index.name = 'id'
    display(df_level)
    return df_level
```

```
28 print("-----非营运车辆-----")
df_no_yingyun_level = get_level(df_no_yingyun_value)
print("-----营运车辆-----")
df_yingyun_level = get_level(df_yingyun_value)
```

-----非营运车辆-----

	age	is_local	trans_code	occur_prob	severity	score	fine
id							
110101197110121532	3.0	2.0	1.0	1.0	1.0	1.0	3.0
11010119720630005X	3.0	1.0	1.0	1.0	1.0	1.0	2.0
110101197211232039	3.0	1.0	1.0	1.0	1.0	1.0	2.0
11010119740316101X	3.0	1.0	4.0	3.0	3.0	1.0	3.0
110102195901261133	4.0	1.0	4.0	1.0	1.0	5.0	3.0
...
654301197712100819	3.0	1.0	1.0	1.0	1.0	1.0	3.0
654323196502241713	4.0	2.0	1.0	4.0	4.0	1.0	4.0
654323198306140017	2.0	2.0	1.0	1.0	1.0	3.0	1.0
659001197008121219	3.0	1.0	1.0	1.0	1.0	1.0	3.0
659001197708260316	3.0	1.0	1.0	3.0	2.0	1.0	3.0

119750 rows × 9 columns

-----营运车辆-----

	age	is_local	trans_code	occur_prob	severity	score	fine
id							
110221197310165610	3.0	2.0	4.0	5.0	4.0	6.0	5.0
110227196610231513	4.0	2.0	3.0	1.0	1.0	1.0	3.0
120105196011093356	4.0	2.0	3.0	3.0	3.0	1.0	5.0
120107197201036931	3.0	2.0	3.0	1.0	1.0	1.0	3.0
120110195702260914	5.0	2.0	3.0	1.0	1.0	1.0	2.0
...
652901196112011413	4.0	2.0	2.0	2.0	5.0	1.0	3.0
653125197802101451	3.0	2.0	3.0	4.0	4.0	1.0	1.0
653126198010212015	2.0	2.0	3.0	1.0	1.0	1.0	2.0
653127198211282016	2.0	1.0	2.0	1.0	1.0	4.0	3.0
813027197712217415	3.0	1.0	2.0	1.0	1.0	1.0	5.0

51771 rows × 9 columns

```
df_no_yingyun_level.loc[df_no_yingyun_level.isin([np.nan]).any(axis=1), 'fine'] = 0
df_no_yingyun_level[df_no_yingyun_level.isin([np.nan]).any(axis=1)]
```

106 # 数据概况

```
import matplotlib.pyplot as plt

fig = plt.figure()
fig.tight_layout()

# 输入的是一个df, 主键是每个司机的驾驶证号id
# 得到这个司机群体的信息
def analyseDF(df_input):
    plt.figure(figsize=(40, 40))

    # 年龄组成
    ages = df_input['age'].value_counts()
    plt.subplot(4, 1, 1)
    df_input['age'].plot(kind='hist')
    plt.title("Count of age")

    # print(ages)
    # 外地or本地
    is_local = df_input['is_local'].value_counts()
    print(is_local)
    # 准驾车型
    trans_codes = df_input['机动车_交通方式说明'].value_counts()
    print(trans_codes)

    type_counts = df_input['违法行为_违法小类'].value_counts()
    print(type_counts.head(10))
    plt.subplot(4, 1, 2)
    word_cloud(type_counts)

    detail_count = df_input['违法行为_违法描述'].value_counts()
    print(detail_count.head(10))
    plt.subplot(4, 1, 3)
    word_cloud(detail_count)

    # 计算违章时间段统计

    df_input['违法时间'] = pd.to_datetime(df_input['违法时间'])
    df_input['违法时间'] = pd.to_datetime(df_input['违法时间'])
    df_date = df_input.sort_values('违法时间')
    df_date = df_date.set_index('违法时间')
    # print(df_date.truncate(before='2019', after='2022-1').head())
    df_date_col = df_input['违法时间']
    df_date_col['hour'] = df_input['违法时间'].dt.hour
    df_date_col['minute'] = df_input['违法时间'].dt.minute
    a = df_date_col['hour'].value_counts().sort_index()
    plt.subplot(4, 1, 4)
    a.plot(kind='bar', x='Time')
    plt.title("Count of Time")
    plt.show()

    # 词云计算
    # 输入变量格式为value_counts的结果
    from wordcloud import WordCloud

    def word_cloud(word_counts):
```

```

counts = {}
for word in word_counts.index:
    counts[word] = word_counts[word]
wordcloud = WordCloud(background_color='white', font_path='/System/Library/Fonts/Hiragino
                      height=1000,
                      margin=2)
wordcloud.generate_from_frequencies(counts)
# 显示图片
plt.imshow(wordcloud)

```

<Figure size 432x288 with 0 Axes>

107 analyseDF(df_no_yingyun_value)
analyseDF(df_yingyun_value)

1	120594
2	37415
Name: is_local, dtype: int64	
小型轿车	67132
轻型栏板货车	35514
重型自卸货车	7898
重型半挂牵引车	7882
轻型厢式货车	4410
轻便二轮摩托车	3985
小型面包车	2963
中型栏板货车	2635
小型普通客车	2504
重型厢式货车	2496
重型非载货专项作业车	2458
小型越野客车	2052
重型仓栅式货车	1620
自卸低速货车	1485
轻型自卸货车	1474
大型轮式拖拉机	1427
中型普通客车	1332
正三轮载货摩托车	1198
小型轮式拖拉机	993
普通二轮摩托车	875
中型厢式货车	609
正三轮载客摩托车	463
轻便正三轮摩托车	442
中型自卸货车	395
重型自卸半挂车	351
微型轿车	306
重型罐式货车	297
手扶拖拉机	261
轻型仓栅式货车	233
微型栏板货车	215
微型面包车	213
三轮汽车	212
微型普通客车	205
普通正三轮摩托车	198
轻型封闭式货车	173
重型仓栅式半挂车	151
重型栏板半挂车	148
小型非载货专项作业车	84
栏板低速货车	76
重型栏板货车	73
重型平板货车	67
手扶变形运输机	46
中型栏板半挂车	43
微型厢式货车	42
轻型栏板半挂车	40
中型非载货专项作业车	36
其它	31
重型专项作业半挂车	30
重型全挂牵引车	25

残疾人专用车	23
重型厢式全挂车	19
重型集装箱半挂车	18
重型罐式半挂车	16
中型半挂牵引车	16
中型罐式货车	14
重型平板半挂车	8
轻型特殊结构货车	7
大型普通客车	6
中型自卸半挂车	6
大型卧铺客车	6
微型自卸货车	6
重型低平板半挂车	6
大型非载货专项作业车	5
中型专用客车	5
重型厢式半挂车	5
重型栏板全挂车	5
中型平板货车	4
轻型栏板全挂车	4
助力自行车	4
轻型平板货车	3
重型封闭式货车	3
轻型自卸半挂车	2
微型封闭式货车	2
轻型厢式半挂车	2
轮式挖掘机械	2
轻型厢式全挂车	2
中型特殊结构货车	2
中型低平板半挂车	1
重型自卸全挂车	1
大型双层客车	1
中型厢式全挂车	1
大型越野客车	1
有轨电车	1
中型厢式半挂车	1
微型越野客车	1
中型罐式半挂车	1
中型自卸全挂车	1

Name: 机动车_交通方式说明, dtype: int64

违法上道路行驶	36778
违反交通信号	27371
违法停车	22919
违法装载	15867
违法变更车道	8501
违法占道行驶	6607
未携带驾驶证	5504
超速行驶	4144
饮酒驾驶	3986
违法抢行	3940

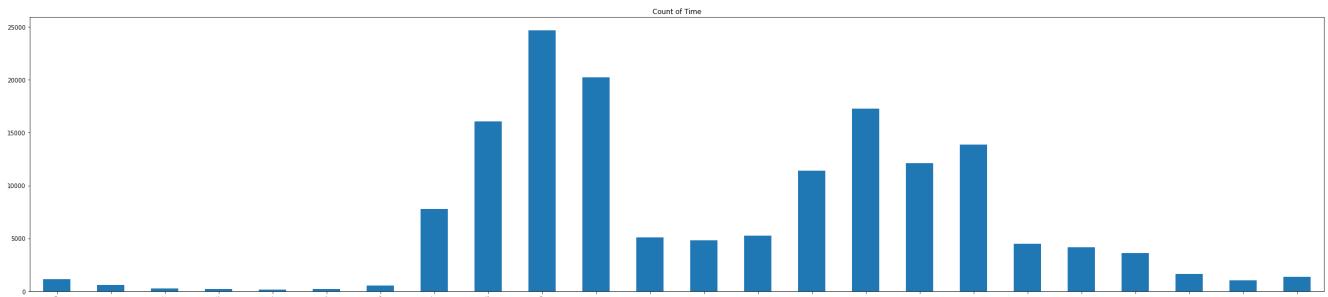
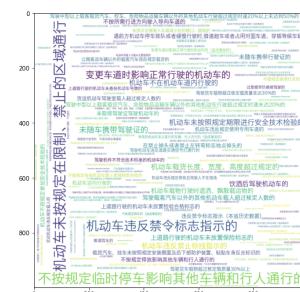
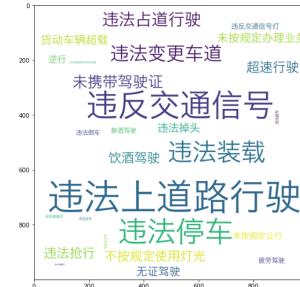
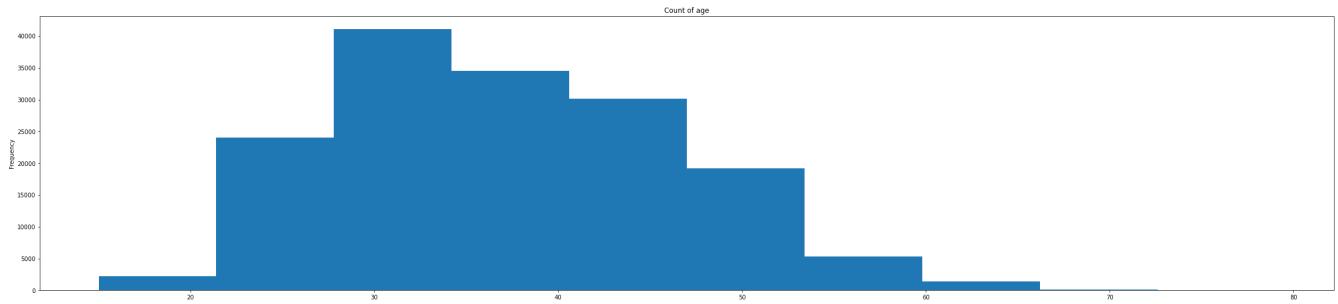
Name: 违法行为_违法小类, dtype: int64

机动车未按规定在限制、禁止的区域通行	20236
不按规定临时停车影响其他车辆和行人通行的	19501
机动车违反禁令标志指示的	17244
变更车道时影响正常行驶的机动车的	7574
未随车携带驾驶证的	5361
机动车未按照规定期限进行安全技术检验的	4534
机动车载货长度、宽度、高度超过规定的	4490
机动车违反禁止标线指示的	4386
机动车不在机动车道内行驶的	4077
饮酒后驾驶机动车的	3852

Name: 违法行为_违法描述, dtype: int64

```
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide  
df_date_col['hour'] = df_input['违法时间'].dt.hour
```



1	83850
2	25043
Name: is_local, dtype: int64	
重型半挂牵引车	29795
重型自卸货车	27899
轻型栏板货车	19724
重型厢式货车	5104
中型栏板货车	4708
小型轿车	3971
重型非载货专项作业车	3451
重型仓栅式货车	2779
轻型厢式货车	2403
轻型自卸货车	1990
中型厢式货车	1084
重型罐式货车	1077
中型自卸货车	1052
中型普通客车	717
自卸低速货车	673
轻型封闭式货车	315
轻型仓栅式货车	199
重型栏板半挂车	169
重型自卸半挂车	161
重型仓栅式半挂车	133
重型平板货车	129
轻便二轮摩托车	123

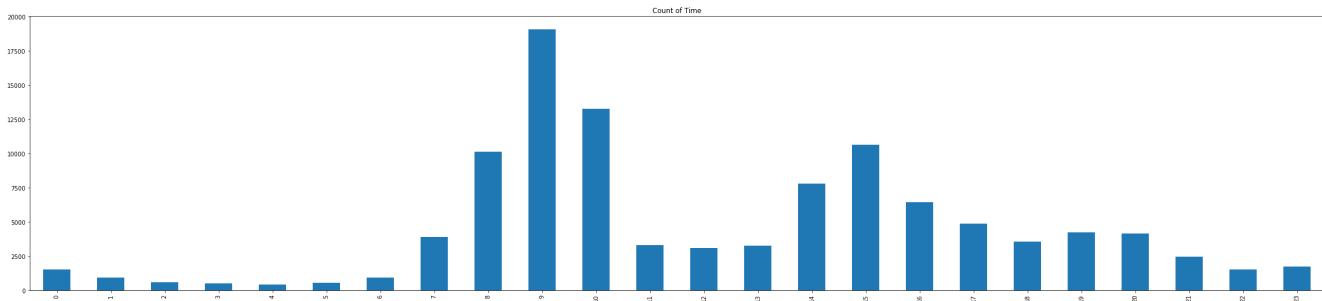
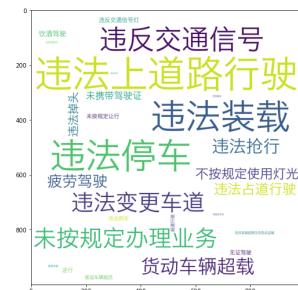
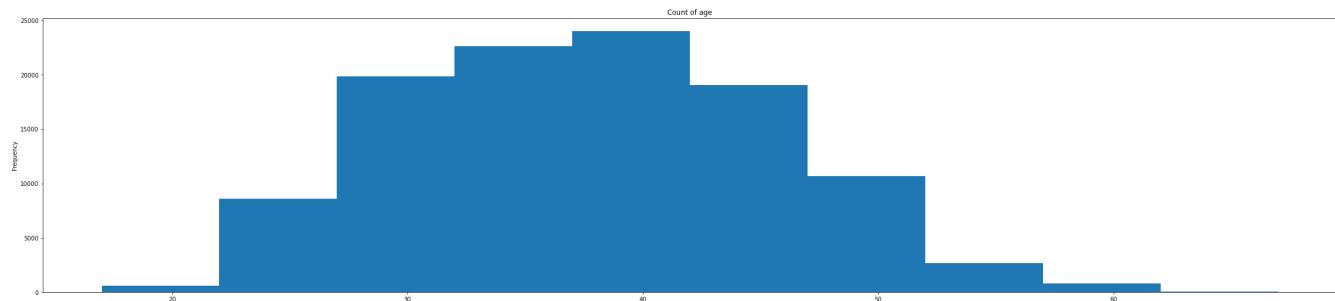
大型轮式拖拉机	115
小型轮式拖拉机	110
手扶拖拉机	110
正三轮载货摩托车	99
重型栏板货车	93
中型罐式货车	89
三轮汽车	70
中型半挂牵引车	45
普通二轮摩托车	44
大型卧铺客车	41
大型普通客车	36
小型面包车	32
重型集装箱半挂车	31
小型普通客车	28
微型栏板货车	28
小型非载货专项作业车	20
重型特殊结构货车	19
小型越野客车	19
中型栏板半挂车	18
栏板低速货车	18
重型封闭式货车	17
中型非载货专项作业车	15
重型罐式半挂车	14
轻型栏板半挂车	14
重型平板半挂车	14
普通正三轮摩托车	10
正三轮载客摩托车	9
中型自卸全挂车	8
重型厢式半挂车	8
微型厢式货车	6
手扶变形运输机	6
重型全挂牵引车	6
重型低平板半挂车	5
微型普通客车	5
大型非载货专项作业车	4
重型厢式全挂车	4
厢式低速货车	4
其它	3
手推车	2
轻型特殊结构货车	2
重型栏板全挂车	2
轻型自卸半挂车	2
微型轿车	2
中型仓栅式货车	2
大型双层客车	2
中型自卸半挂车	1
中型集装箱半挂车	1
中型特殊结构货车	1
中型平板货车	1
中型低平板半挂车	1
重型专项作业半挂车	1
Name: 机动车_交通方式说明, dtype: int64	
违法上道路行驶	22236
违法停车	19743
违法装载	19036
违反交通信号	10870
未按规定办理业务	7436
违法变更车道	6923
货运车辆超载	4812
违法抢行	3817
疲劳驾驶	2978
违法占道行驶	2470
Name: 违法行为_违法小类, dtype: int64	
不按规定临时停车影响其他车辆和行人通行的	17235
机动车未按规定在限制、禁止的区域通行	16091
机动车违反禁令标志指示的	7352
机动车载物行驶时遗洒、飘散载运物的	7237
载货汽车、挂车未按照规定安装侧面及后下部防护装置、粘贴车身反光标识的	7200

```

变更车道时影响正常行驶的机动车的 5406
机动车载货长度、宽度、高度超过规定的 4346
未依次交替驶入车道减少后的路口、路段的 3178
过度疲劳仍继续驾驶的 2976
驾驶货车载物超过核定载质量30%以上 2574
Name: 违法行为_违法描述, dtype: int64
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html#inplace



聚类操作

```

80 # k-mean聚类
# 数据标准化后的聚类计算
from sklearn.cluster import KMeans

```

```

def Kmeans(df_input, list_variables):
    X = df_input[list_variables].values
    k_means = KMeans(n_clusters=4, random_state=0).fit(X)

```

```

df_input['label'] = k_means.labels_
print('聚类的个数')
print(df_input['label'].value_counts())

label_pred = k_means.labels_
centroids = k_means.cluster_centers_
inertia = k_means.inertia_
length, dim = X.shape
df_center = pd.DataFrame(centroids, columns=list_variables)
display(df_center)
return df_input

def choose_k(df_input, list_variables):
    SSE = [] # 存放每次结果的误差平方和
    for k in range(1, 9):
        estimator = KMeans(n_clusters=k) # 构造聚类器
        estimator.fit(df_input[list_variables])
        SSE.append(estimator.inertia_)
    X = range(1, 9)
    plt.xlabel('k')
    plt.ylabel('SSE')
    plt.plot(X, SSE, 'o-')
    plt.show()

```

86 print(df_no_yingyun_level.columns)

```

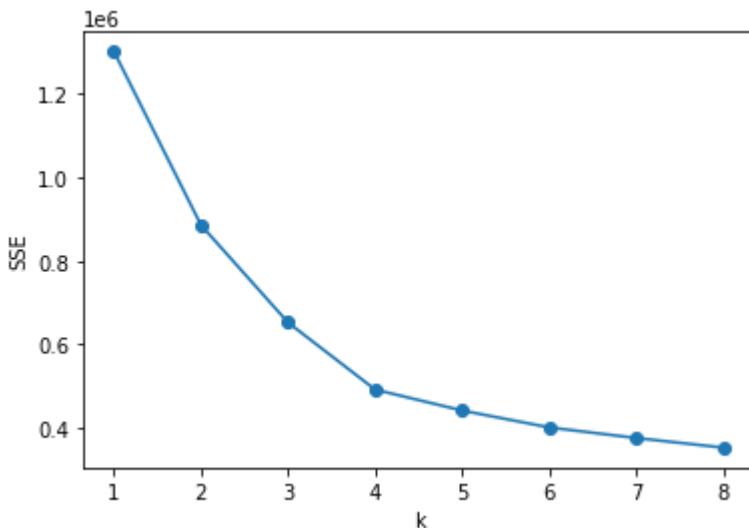
Index(['age', 'is_local', 'trans_code', 'occur_prob', 'severity', 'score',
       'fine', 'accident', 'counts', 'label'],
      dtype='object')

```

非营运车辆结果

87 list_variables = ['age', 'is_local', 'trans_code', 'occur_prob', 'severity', 'score',
 'fine', 'accident', 'counts']

78 choose_k(df_no_yingyun_level, list_variables)



81 cluster_no_yingyun = Kmeans(df_no_yingyun_level, list_variables)

聚类的个数

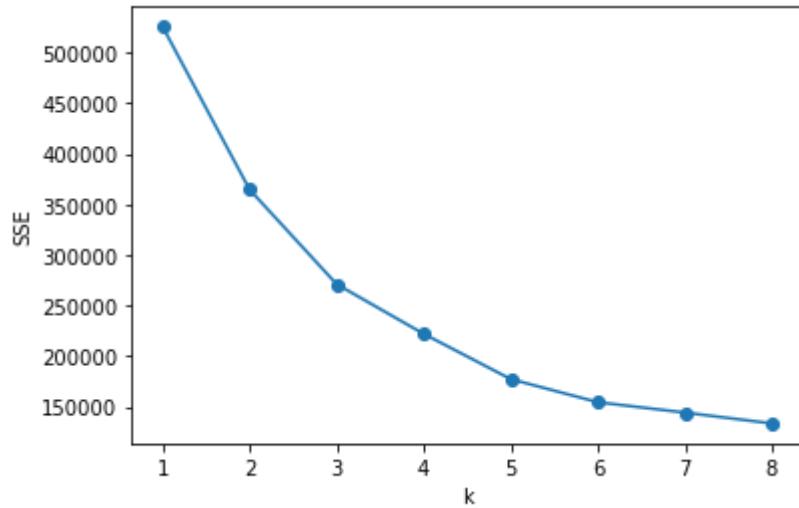
1 38052

```
3    36649  
2    30810  
0    14239  
Name: label, dtype: int64
```

	age	is_local	trans_code	occur_prob	severity	score	accident	
0	2.789507	1.176851	1.636044	4.323992	4.296390	5.275460	1.017067	1.1
1	2.771733	1.230632	1.728687	1.064228	1.072611	1.000000	1.008357	1.2
2	2.784680	1.233853	1.943914	3.172769	3.935670	1.262188	1.015677	1.3
3	2.732005	1.176426	1.806548	1.112879	1.125048	4.437599	1.012115	1.3

营运车辆结果

```
79 choose_k(df_yingyun_level, list_variables)
```



```
82 cluster_yingyun = Kmeans(df_yingyun_level, list_variables)
```

聚类的个数

```
3    16306  
0    15692  
1    12656  
2    7117  
Name: label, dtype: int64
```

	age	is_local	trans_code	occur_prob	severity	score	accident	
0	2.861233	1.410624	2.364326	3.064664	4.322556	1.364262	1.010267	1.2
1	2.802845	1.227262	2.229633	1.728487	2.377953	4.966495	1.014777	1.3
2	2.752981	1.184598	2.287137	1.865900	2.357554	1.778370	1.003647	4.1
3	2.795535	1.252913	2.280388	1.095548	1.067399	1.082792	1.005397	1.2

```
54 cluster_no_yingyun.to_csv("/Users/jiarui/Study/交通事故/data/output/df_no_yingyun_cluster_output.csv")  
cluster_yingyun.to_csv("/Users/jiarui/Study/交通事故/data/output/df_yingyun_cluster_output.csv")
```

```

108 def analyse_clusters(df_cluster, df_all_info):
    for i in range(0, 4):
        print('-----k = {}-----'.format(i))
        dfi = df_cluster[df_cluster['label'] == i]
        list_ids = list(dfi.index)
        print('共有违法者{}人'.format(len(list_ids)))
        list_analyse = ['当事人_驾驶证号', '机动车_交通方式说明', '违法行为_违法小类', '机动车_号牌号码',
                        'is_local',
                        'age', 'severity']
        df_analyse = df_all_info[df_all_info['当事人_驾驶证号'].isin(list_ids)][list_analyse]
        print('共有违法记录{}条'.format(len(df_analyse)))
        df_analyse['机动车_号牌号码'] = df_analyse['机动车_号牌号码'].str[0:2]
        analyseDF(df_analyse)

```

非营运

109 analyse_clusters(cluster_no_yingyun, df_no_yingyun_value)

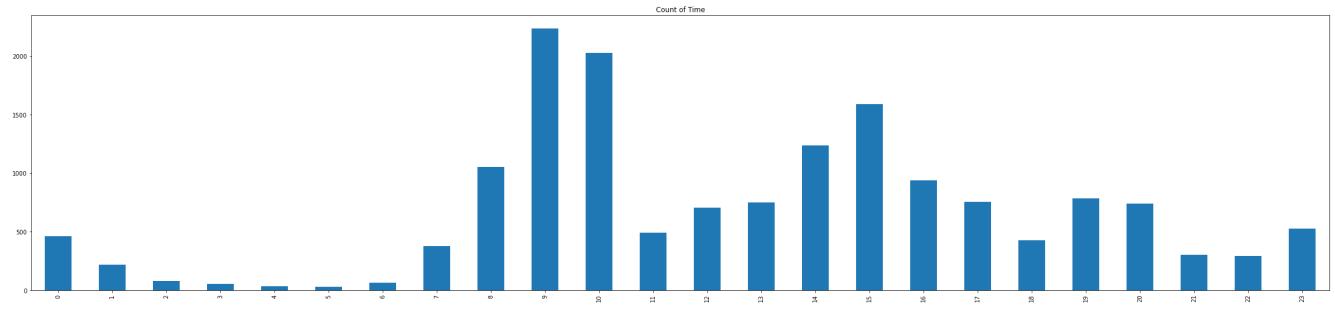
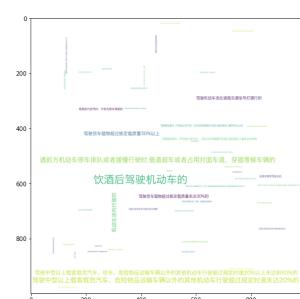
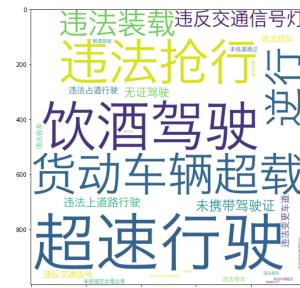
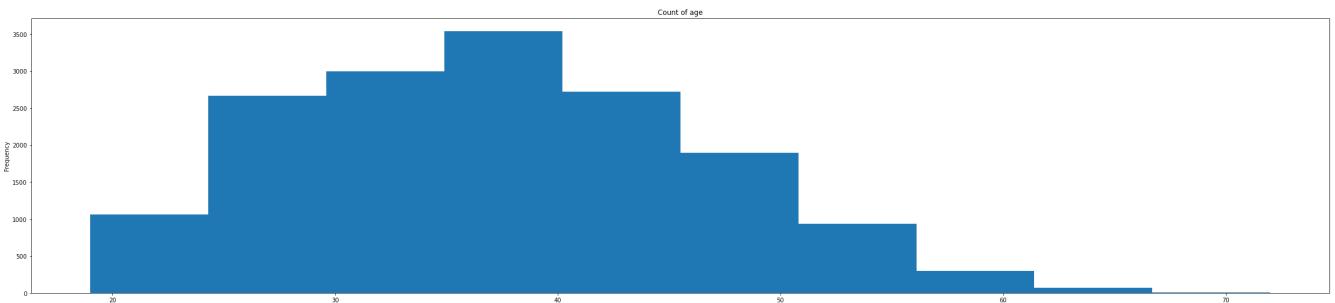
-----k = 0-----

共有违法者14239人

共有违法记录16200条

1	13197
2	3003
Name: is_local, dtype: int64	
小型轿车	9451
轻型栏板货车	2859
小型面包车	389
轻便二轮摩托车	385
小型普通客车	375
重型自卸货车	350
小型越野客车	336
重型半挂牵引车	330
轻型厢式货车	260
中型普通客车	166
轻型自卸货车	148
中型栏板货车	125
正三轮载货摩托车	109
自卸低速货车	109
重型厢式货车	105
大型轮式拖拉机	85
小型轮式拖拉机	84
普通二轮摩托车	80
重型非载货专项作业车	63
重型仓栅式货车	44
中型自卸货车	44
中型厢式货车	28
手扶拖拉机	23
微型轿车	23
微型栏板货车	23
重型罐式货车	22
正三轮载客摩托车	21
三轮汽车	19
微型普通客车	18
微型面包车	15
普通正三轮摩托车	13
轻型封闭式货车	12
小型非载货专项作业车	12
轻型仓栅式货车	10
重型平板货车	9
重型自卸半挂车	8
轻便正三轮摩托车	7
栏板低速货车	7
中型栏板半挂车	4
重型仓栅式半挂车	4
重型栏板货车	4

微型厢式货车	4
轻型栏板半挂车	3
手扶变形运输机	3
重型罐式半挂车	2
重型栏板半挂车	2
中型自卸半挂车	1
重型全挂牵引车	1
大型普通客车	1
中型厢式全挂车	1
微型越野客车	1
重型专项作业半挂车	1
重型集装箱半挂车	1
Name: 机动车_交通方式说明, dtype: int64	
超速行驶	3885
饮酒驾驶	2797
违法抢行	2327
货动车辆超载	1983
逆行	1540
违法装载	950
违反交通信号灯	869
未携带驾驶证	358
无证驾驶	257
违法上道路行驶	249
Name: 违法行为_违法小类, dtype: int64	
饮酒后驾驶机动车的	2769
驾驶中型以上载客载货汽车、危险物品运输车辆以外的其他机动车行驶超过规定时速未达20%的	1954
遇前方机动车停车排队或者缓慢行驶时,借道超车或者占用对面车道、穿插等候车辆的	1894
驾驶中型以上载客载货汽车、校车、危险物品运输车辆以外的其他机动车行驶超过规定时速20%以上未达到50%的	1756
机动车逆向行驶的	1538
驾驶货车载物超过核定载质量30%以上	1155
驾驶机动车违反道路交通信号灯通行的	869
驾驶货车载物超过核定载质量未达30%的	825
等候放行信号时,不依次停车等候的	403
未随车携带驾驶证的	326
Name: 违法行为_违法描述, dtype: int64	
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:	
A value is trying to be set on a copy of a slice from a DataFrame	
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide	
df_date_col['hour'] = df_input['违法时间'].dt.hour	



-----k = 1-----

共有违法者38052人

共有违法记录49143条

1 36515

2 12628

Name: is_local, dtype: int64

小型轿车 22696

轻型栏板货车 8811

重型半挂牵引车 3336

重型自卸货车 3242

轻型厢式货车 1111

重型厢式货车 977

中型栏板货车 845

重型非载货专项作业车 782

小型面包车 718

小型普通客车 702

重型仓栅式货车 669

轻便二轮摩托车 663

小型越野客车 659

自卸低速货车 554

轻型自卸货车 457

大型轮式拖拉机 353

轻便正三轮摩托车 296

小型轮式拖拉机 248

中型普通客车	209
中型厢式货车	203
正三轮载货摩托车	193
普通二轮摩托车	137
重型罐式货车	127
重型自卸半挂车	126
中型自卸货车	111
微型轿车	97
重型仓栅式半挂车	73
正三轮载客摩托车	72
手扶拖拉机	66
重型栏板半挂车	64
轻型仓栅式货车	51
普通正三轮摩托车	46
轻型封闭式货车	43
微型栏板货车	42
微型普通客车	36
小型非载货专项作业车	34
三轮汽车	30
重型栏板货车	22
微型面包车	21
其它	18
轻型栏板半挂车	17
中型栏板半挂车	16
重型全挂牵引车	16
重型专项作业半挂车	14
栏板低速货车	14
重型平板货车	12
重型厢式全挂车	12
残疾人专用机动车	11
手扶变形运输机	11
中型罐式货车	10
微型厢式货车	8
中型半挂牵引车	7
重型罐式半挂车	6
重型集装箱半挂车	5
中型非载货专项作业车	5
大型卧铺客车	4
大型普通客车	4
大型非载货专项作业车	3
微型自卸货车	3
重型平板半挂车	3
轮式挖掘机械	2
重型厢式半挂车	2
轻型自卸半挂车	2
中型特殊结构货车	2
重型封闭式货车	2
微型封闭式货车	2
重型低平板半挂车	1
中型平板货车	1
大型越野客车	1
中型自卸半挂车	1
助力自行车	1
中型自卸全挂车	1
中型厢式半挂车	1
中型低平板半挂车	1
轻型特殊结构货车	1
轻型厢式半挂车	1
Name: 机动车_交通方式说明, dtype: int64	
违法上道路行驶	21670
违法停车	20009
未按规定办理业务	1576
违反交通信号	1285
未携带驾驶证	853
无证驾驶	668
违法倒车	627
违法占道行驶	572
违法装载	455

违法掉头 351

Name: 违法行为_违法小类, dtype: int64

17802

不按规定临时停车影响其他车辆和行人通行的

17082

不按规定停放影响其他车辆和行人通行的，对驾驶人处以罚款一百元，记三分。

1913

载货汽车、挂车未按照规定安装侧面及后下部防护装置、粘贴车身反光标识的

1207

驾驶机件不符合技术标准的机动车的

1056

未随车携带驾驶证的

3

机动车违反规定停放且驾驶人不在现场上道路行驶的机动车未放置保险标志的

132

上道路行驶的机动车未放置保险标志的，由公安机关交通管理部门予以扣留机动车，

624

未取得驾驶证驾驶机动车的
由公安机关进行处罚

621

未随车携带行驶证的
违法行驶

58

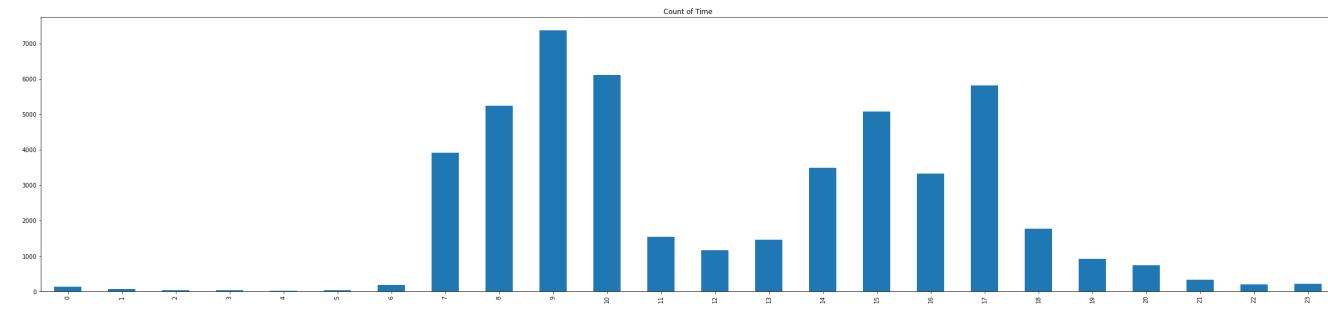
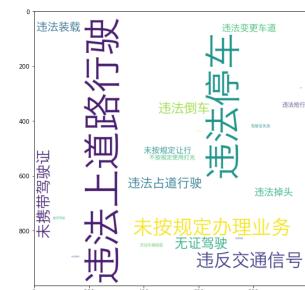
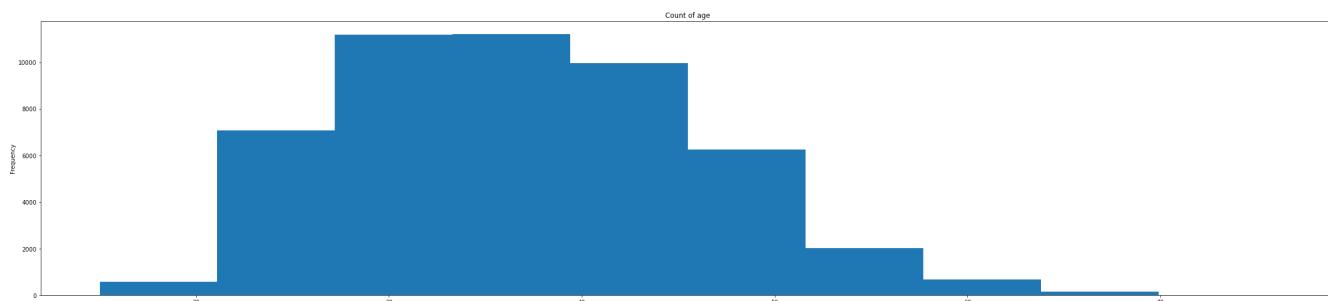
Name: 违法行为_违法描述, dtype: int64

```
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/datetime.html

```
df_date_col['hour'] = df_input['违法时间'].dt.hour
```



———
从古诗谈起

共有违法者30810人

共有违法记录

1 29595

Name: is_local, dtype: int64

小型轿车	13775
轻型栏板货车	9555
重型自卸货车	3526
重型半挂牵引车	2160
轻便二轮摩托车	2123
重型非载货专项作业车	1271
小型面包车	1054
轻型厢式货车	1008
中型栏板货车	950
小型普通客车	817
自卸低速货车	610
中型普通客车	608
大型轮式拖拉机	606
轻型自卸货车	514
小型轮式拖拉机	451
重型厢式货车	439
小型越野客车	383
重型仓栅式货车	277
正三轮载货摩托车	224
重型自卸半挂车	187
中型自卸货车	167
中型厢式货车	125
普通二轮摩托车	111
重型罐式货车	93
微型面包车	89
手扶拖拉机	84
微型普通客车	82
轻便正三轮摩托车	73
轻型仓栅式货车	72
重型栏板半挂车	50
三轮汽车	45
轻型封闭式货车	42
微型轿车	41
正三轮载客摩托车	37
栏板低速货车	36
微型栏板货车	33
重型仓栅式半挂车	33
重型栏板货车	31
普通正三轮摩托车	28
手扶变形运输机	20
中型栏板半挂车	14
小型非载货专项作业车	14
残疾人专用汽车	12
重型集装箱半挂车	12
微型厢式货车	11
重型平板货车	11
轻型栏板半挂车	10
中型非载货专项作业车	10
其它	9
重型全挂牵引车	7
轻型特殊结构货车	6
重型罐式半挂车	5
重型栏板全挂车	4
重型厢式全挂车	4
中型自卸半挂车	4
中型半挂牵引车	4
中型平板货车	3
助力自行车	3
重型平板半挂车	3
微型自卸货车	2
大型非载货专项作业车	2
重型专项作业半挂车	2
轻型栏板全挂车	2
中型专用客车	2
重型低平板半挂车	2
大型卧铺客车	2
中型罐式货车	2

中型罐式半挂车 1
重型自卸全挂车 1
大型双层客车 1
有轨电车 1
重型厢式半挂车 1
轻型平板货车 1

Name: 机动车_交通方式说明, dtype: int64

违法装载 13648
违法变更车道 7586
违法占道行驶 5461
不按规定使用灯光 3576
违法掉头 2043
违法上道路行驶 1622
无证驾驶 1543
违法停车 1286
违法抢行 1130
饮酒驾驶 945

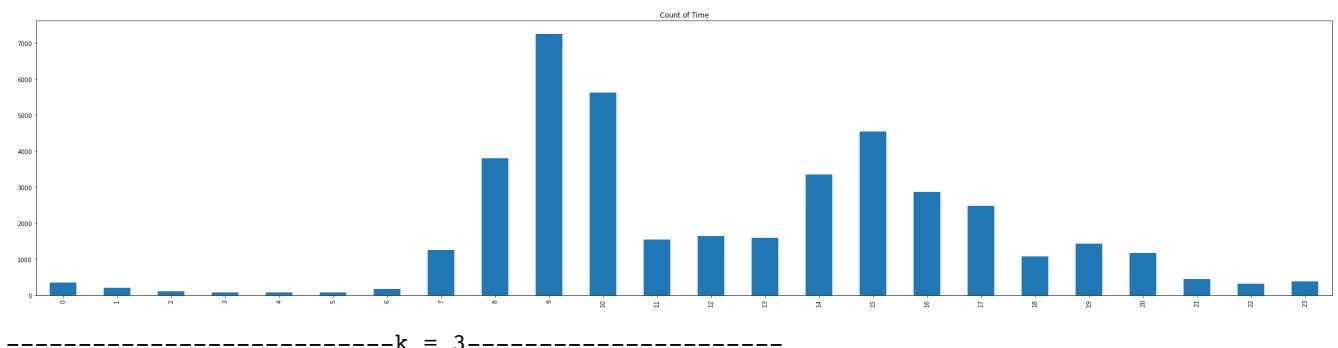
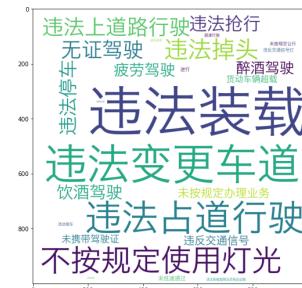
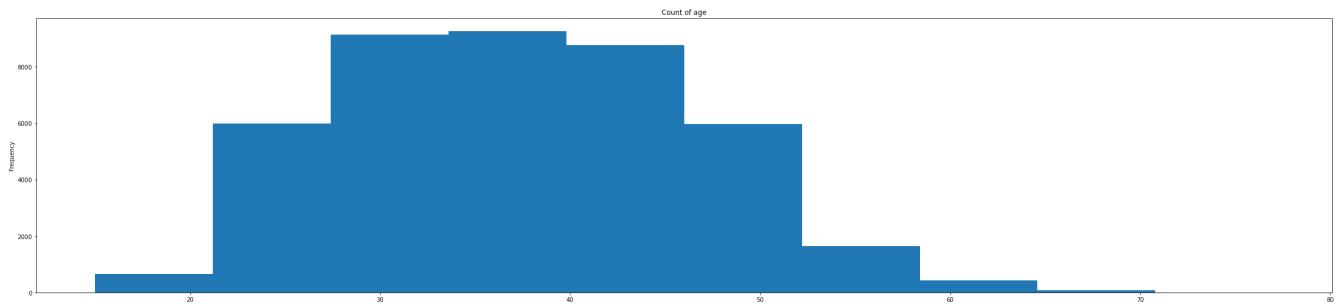
Name: 违法行为_违法小类, dtype: int64

变更车道时影响正常行驶的机动车的 6730
机动车载货长度、宽度、高度超过规定的 3969
机动车不在机动车道内行驶的 3451
机动车载物行驶时遗洒、飘散载运物的 2955
驾驶载客汽车以外的其他机动车载人超过核定人数的 2661
机动车不按规定参加安全技术检验的 2201
在禁止掉头或者禁止左转弯标志地点掉头的 1700
货运机动车驾驶室载人超过核定人数的 1633
机动车违反规定使用专用车道的 1454
未取得驾驶证驾驶机动车的 1395

Name: 违法行为_违法描述, dtype: int64

<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
df_date_col['hour'] = df_input['违法时间'].dt.hour



中型厢式货车	253
自卸低速货车	212
小型轮式拖拉机	210
微型轿车	145
三轮汽车	118
微型栏板货车	117
普通正三轮摩托车	111
轻型仓栅式货车	100
微型面包车	88
手扶拖拉机	88
轻型封闭式货车	76
中型自卸货车	73
微型普通客车	69
轻便正三轮摩托车	66
重型罐式货车	55
重型仓栅式半挂车	41
重型平板货车	35
重型栏板半挂车	32
重型自卸半挂车	30
小型非载货专项作业车	24
中型非载货专项作业车	21
微型厢式货车	19
栏板低速货车	19
重型栏板货车	16
重型专项作业半挂车	13
手扶变形运输机	12
轻型栏板半挂车	10
中型栏板半挂车	9
中型半挂牵引车	5
其它	4
重型厢式全挂车	3
重型低平板半挂车	3
中型专用客车	3
重型罐式半挂车	3
轻型厢式全挂车	2
中型罐式货车	2
重型平板半挂车	2
重型厢式半挂车	2
轻型栏板全挂车	2
轻型平板货车	2
微型自卸货车	1
重型全挂牵引车	1
轻型厢式半挂车	1
重型栏板全挂车	1
大型普通客车	1
重型封闭式货车	1

Name: 机动车_交通方式说明, dtype: int64

违反交通信号	25578
违法上道路行驶	13237
未携带驾驶证	4007
违法停车	1561
未按规定让行	1155
未按规定办理业务	935
违法装载	814
货运车辆超载	675
违法变更车道	443
违法占道行驶	441

Name: 违法行为_违法小类, dtype: int64

机动车违反禁令标志指示的	16431
机动车未按照规定期限进行安全技术检验的	4275
机动车违反禁止标线指示的	4213
未随车携带驾驶证的	3957
上道路行驶的机动车未放置保险标志的	2542
不按所需行进方向驶入导向车道的	2510
未随车携带行驶证的	2243
违反禁令标志指示 (本省历史数据)	1999
机动车未按规定在限制、禁止的区域通行	1671

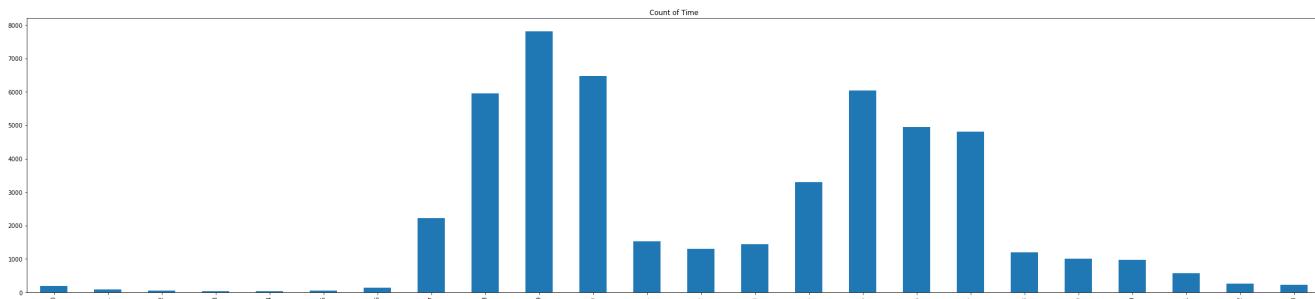
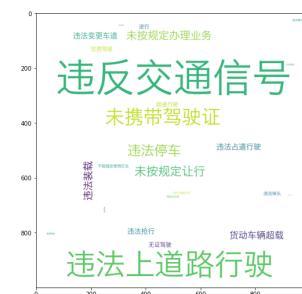
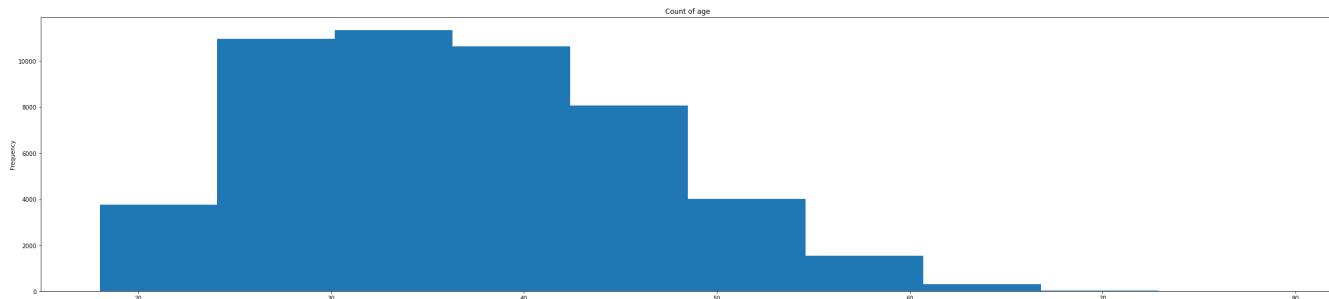
上道路行驶的机动车未放置检验合格标志的 1651

1651

Name: 违法行为_违法描述, dtype: int64

```
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/dt.html



营运

```
110 analyse_clusters(cluster_yingyun, df_yingyun_value)
```

$k = 0$

共有违法者15692人

共有违法记录19992条

1 12086

2 7906

Name: is 1.

重型半挂牵引

轻型栏板货车

重型自卸货车

重型厢式货车

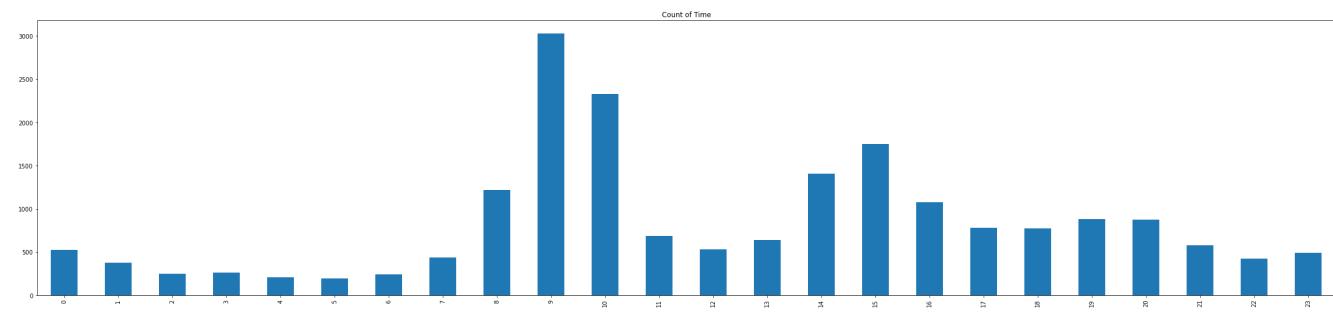
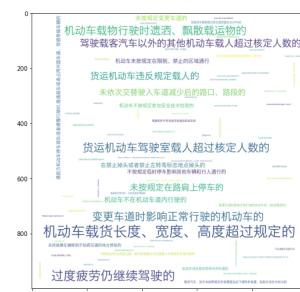
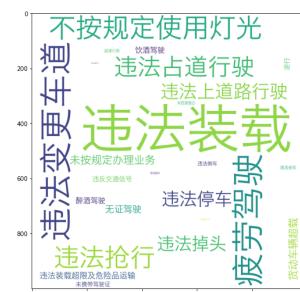
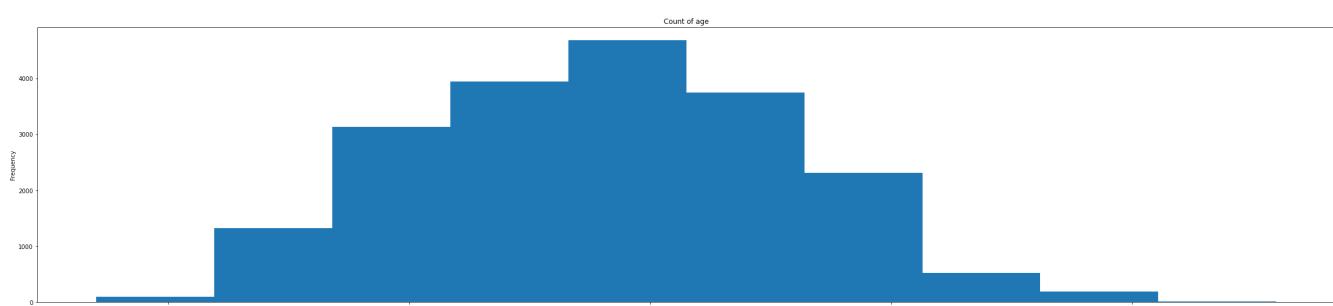
中型栏板货车	1046
小型轿车	845
重型仓栅式货车	593
轻型厢式货车	413
轻型自卸货车	324
重型非载货专项作业车	280
中型自卸货车	184
中型厢式货车	154
自卸低速货车	132
重型罐式货车	110
中型普通客车	96
轻型仓栅式货车	57
重型栏板半挂车	56
重型仓栅式半挂车	51
小型轮式拖拉机	38
手扶拖拉机	29
重型自卸半挂车	29
大型轮式拖拉机	21
中型罐式货车	20
轻型封闭式货车	17
重型栏板货车	15
中型半挂牵引车	15
重型平板货车	13
重型集装箱半挂车	10
重型平板半挂车	10
中型栏板半挂车	10
三轮汽车	9
轻型栏板半挂车	8
栏板低速货车	8
微型栏板货车	8
大型卧铺客车	6
小型越野客车	4
正三轮载货摩托车	4
重型罐式半挂车	4
重型封闭式货车	3
手扶变形运输机	3
重型厢式全挂车	3
轻便二轮摩托车	3
普通二轮摩托车	3
微型厢式货车	3
小型非载货专项作业车	3
重型厢式半挂车	2
大型普通客车	2
小型面包车	2
轻型自卸半挂车	2
重型全挂牵引车	2
中型非载货专项作业车	2
重型栏板全挂车	2
小型普通客车	1
其它	1
中型自卸半挂车	1
普通正三轮摩托车	1
重型特殊结构货车	1
重型低平板半挂车	1
正三轮载客摩托车	1
手推车	1
中型低平板半挂车	1

Name: 机动车_交通方式说明, dtype: int64	
违法装载	9141
违法变更车道	2350
疲劳驾驶	2334
不按规定使用灯光	1422
违法抢行	1262
违法占道行驶	1065
违法上道路行驶	591
违法停车	556
违法掉头	545
未按规定办理业务	135

Name: 违法行为_违法小类, dtype: int64	
机动车载货长度、宽度、高度超过规定的	3008
过度疲劳仍继续驾驶的	2334
货运机动车驾驶室载人超过核定人数的	1816
变更车道时影响正常行驶的机动车的	1733
机动车载物行驶时遗洒、飘散载运物的	1694
驾驶载客汽车以外的其他机动车载人超过核定人数的	1389
货运机动车违反规定载人的	1211
未按规定在路肩上停车的	1099
未依次交替驶入车道减少后的路口、路段的	909
机动车不在机动车道内行驶的	603
违法行头、违法掉头	1

```
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/datetime.html#date-column



共有违法者12656人

共有违法记录
1 13538

2 3871

轻型栏板货车	5087
重型半挂牵引车	3875
重型自卸货车	1690
小型轿车	1292
重型厢式货车	1033
中型栏板货车	874
轻型厢式货车	705
重型仓栅式货车	502
轻型自卸货车	475
中型普通客车	356
中型厢式货车	313
重型非载货专项作业车	195
中型自卸货车	163
自卸低速货车	148
重型罐式货车	144
轻型仓栅式货车	74
正三轮载货摩托车	44
重型平板货车	39
三轮汽车	38
轻型封闭式货车	28
大型轮式拖拉机	26
普通二轮摩托车	25
中型罐式货车	20
大型普通客车	18
重型仓栅式半挂车	18
小型轮式拖拉机	17
小型普通客车	17
轻便二轮摩托车	16
重型栏板货车	16
微型栏板货车	15
小型面包车	15
大型卧铺客车	15
重型自卸半挂车	13
重型集装箱半挂车	12
中型非载货专项作业车	11
重型栏板半挂车	10
小型越野客车	10
手扶拖拉机	10
中型半挂牵引车	7
普通正三轮摩托车	6
重型罐式半挂车	5
正三轮载客摩托车	5
栏板低速货车	4
中型栏板半挂车	4
微型普通客车	4
重型封闭式货车	2
轻型栏板半挂车	2
大型双层客车	1
重型特殊结构货车	1
重型厢式全挂车	1
小型非载货专项作业车	1
中型平板货车	1
其它	1
重型低平板半挂车	1
微型厢式货车	1
手扶变形运输机	1
中型特殊结构货车	1
轻型特殊结构货车	1

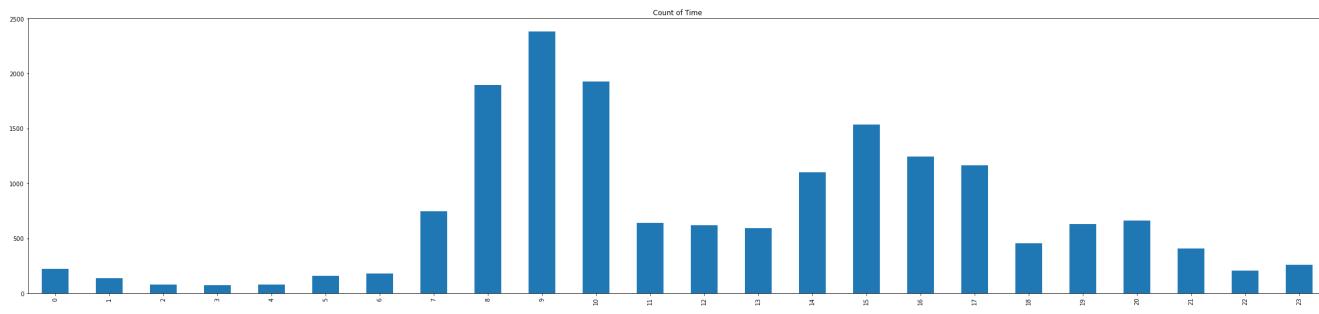
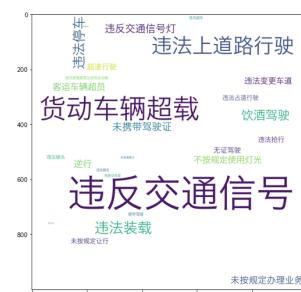
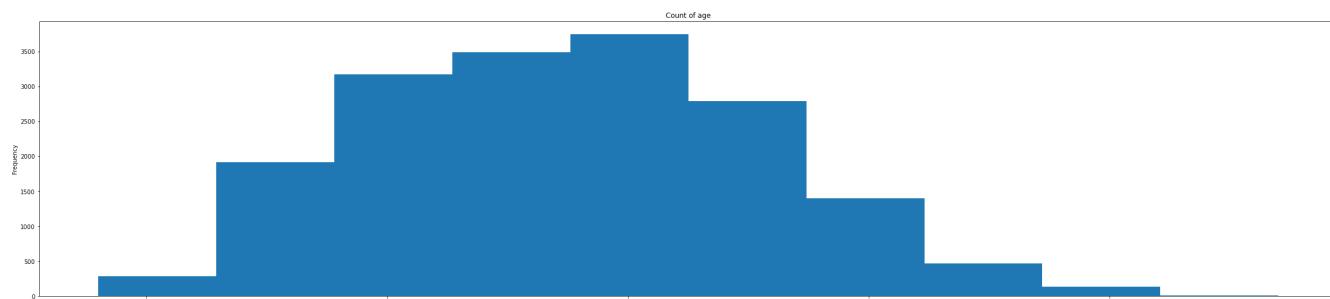
Name: 机动车_交通方式说明, dtype: int64

违反交通信号	7895
货动车辆超载	3250
违法上道路行驶	1689
违法装载	676
违法停车	560
饮酒驾驶	465
违反交通信号灯	322
未携带驾驶证	312
逆行	266

未按规定办理业务 263
Name: 违法行为_违法小类, dtype: int64
机动车违反禁令标志指示的 5638
驾驶货车载物超过核定载质量30%以上 2065
违反禁令标志指示 (本省历史数据) 1639
驾驶货车载物超过核定载质量未达30%的 1177
机动车未按规定在限制、禁止的区域通行 678
机动车未按照规定期限进行安全技术检验的 503
饮酒后驾驶机动车的 456
不按规定临时停车影响其他车辆和行人通行的 434
驾驶机动车违反道路交通信号灯通行的 322
机动车载货长度、宽度、高度超过规定的 310
Name: 违法行为_违法描述, dtype: int64

```
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html#inplace
df_date_col['hour'] = df_input['违法时间'].dt.hour



-----k = 2-----

共有违法者7117人
共有违法记录50699条

1 42442
2 8257

Name: is_local, dtype: int64

重型自卸货车	22312
重型半挂牵引车	12362
轻型栏板货车	4964
重型非载货专项作业车	2573
中型栏板货车	1720
重型厢式货车	1662
重型仓栅式货车	942
轻型自卸货车	808
重型罐式货车	675
轻型厢式货车	567
中型自卸货车	548
自卸低速货车	266
中型厢式货车	260
小型轿车	238
轻型封闭式货车	181
重型自卸半挂车	81
重型平板货车	51
大型轮式拖拉机	50
重型栏板半挂车	50
轻便二轮摩托车	46
正三轮载货摩托车	37
重型栏板货车	37
轻型仓栅式货车	35
中型普通客车	34
重型仓栅式半挂车	30
中型罐式货车	28
小型轮式拖拉机	23
手扶拖拉机	22
重型特殊结构货车	17
三轮汽车	15
普通二轮摩托车	14
中型半挂牵引车	14
重型封闭式货车	12
中型自卸全挂车	8
小型非载货专项作业车	7
厢式低速货车	4
栏板低速货车	3
重型集装箱半挂车	1
轻型栏板半挂车	1
中型栏板半挂车	1

Name: 机动车_交通方式说明, dtype: int64

违法停车	11786
违法上道路行驶	9402
违法装载	9219
未按规定办理业务	5748
违法变更车道	4018
违反交通信号	2528
违法抢行	2302
货动车辆超载	1435
违法占道行驶	1135
违法掉头	975

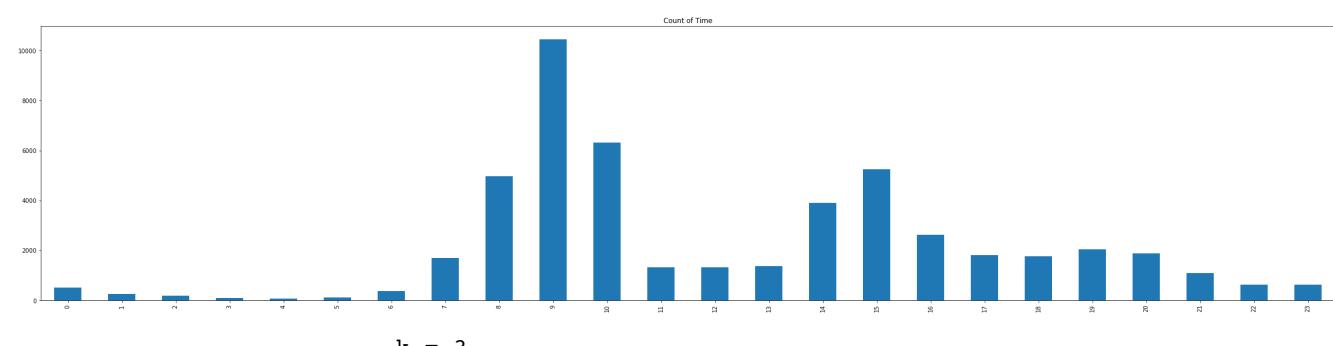
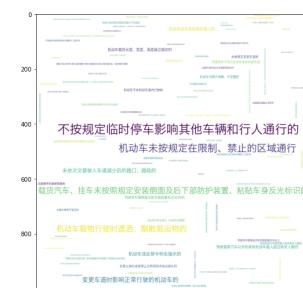
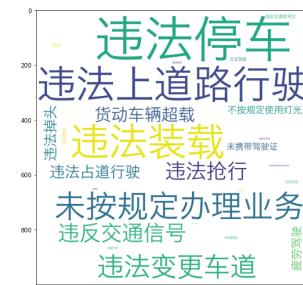
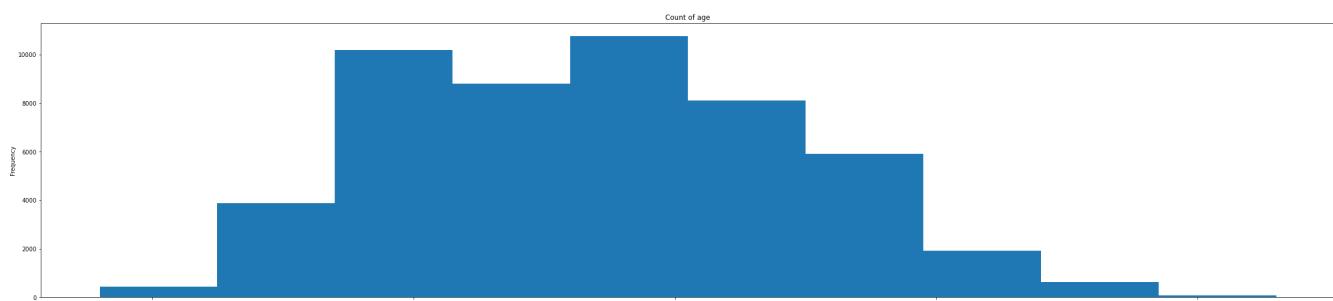
Name: 违法行为_违法小类, dtype: int64

不按规定临时停车影响其他车辆和行人通行的	10898
机动车未按规定在限制、禁止的区域通行	6784
载货汽车、挂车未按照规定安装侧面及后下部防护装置、粘贴车身反光标识的	5647
机动车载物行驶时遗洒、飘散载运物的	5431
变更车道时影响正常行驶的机动车的	3183
未依次交替驶入车道减少后的路口、路段的	2117
机动车违反禁令标志指示的	1583
货运机动车违反规定载人的	1237
驾驶载客汽车以外的其他机动车载人超过核定人数的	1045
机动车载货长度、宽度、高度超过规定的	1028

Name: 违法行为_违法描述, dtype: int64

```
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html
df_date_col['hour'] = df_input['违法时间'].dt.hour



共有违法者16306人

共有违法记录20793条

1 15784

2 5009

Name: is_local, dtype: int64

重型半挂牵引车 6083

轻型栏板货车 4726

重型自卸货车 2119

小型轿车 1596

重型厢式货车 1280

中型栏板货车 1068

重型仓栅式货车 742

轻型厢式货车 718

重型非载货专项作业车 403

轻型自卸货车 383

中型厢式货车 357

中型普通客车 231

中型自卸货车 157

重型罐式货车 148

自卸低速货车 127

轻型封闭式货车 89

轻便二轮摩托车	58
重型栏板半挂车	53
手扶拖拉机	49
重型自卸半挂车	38
重型仓栅式半挂车	34
轻型仓栅式货车	33
小型轮式拖拉机	32
重型平板货车	26
重型栏板货车	25
中型罐式货车	21
大型卧铺客车	20
大型轮式拖拉机	18
大型普通客车	16
小型面包车	15
正三轮载货摩托车	14
小型普通客车	10
小型非载货专项作业车	9
中型半挂牵引车	9
重型集装箱半挂车	8
三轮汽车	8
重型厢式半挂车	6
微型栏板货车	5
小型越野客车	5
重型罐式半挂车	5
大型非载货专项作业车	4
重型平板半挂车	4
重型全挂牵引车	4
正三轮载客摩托车	3
重型低平板半挂车	3
中型栏板半挂车	3
普通正三轮摩托车	3
栏板低速货车	3
轻型栏板半挂车	3
手扶变形运输机	2
中型非载货专项作业车	2
中型仓栅式货车	2
微型轿车	2
微型厢式货车	2
普通二轮摩托车	2
手推车	1
重型专项作业半挂车	1
微型普通客车	1
其它	1
大型双层客车	1
中型集装箱半挂车	1
轻型特殊结构货车	1

Name: 机动车_交通方式说明, dtype: int64

违法上道路行驶	10554
违法停车	6841
未按规定办理业务	1290
未携带驾驶证	439
违反交通信号	402
违法变更车道	344
违法倒车	208
违法占道行驶	148
不按规定使用灯光	144
违法抢行	114

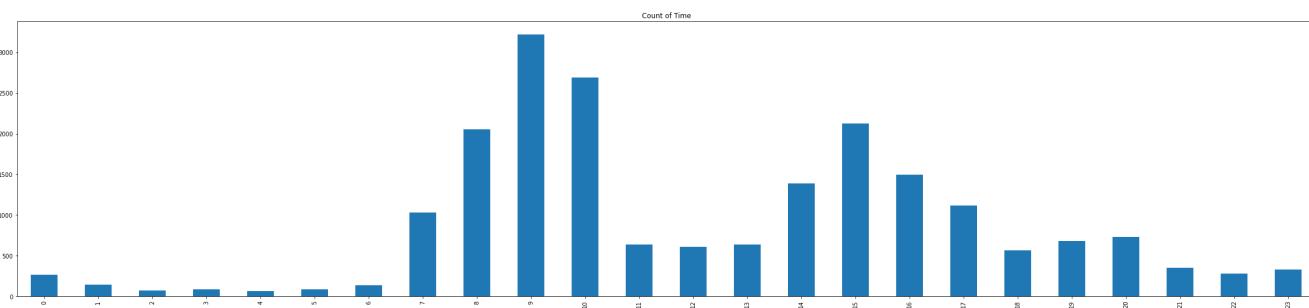
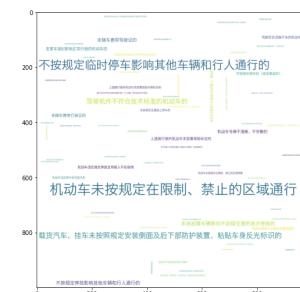
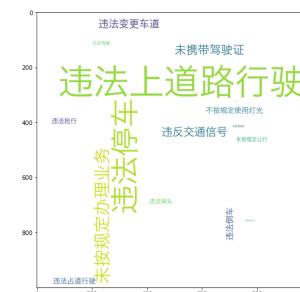
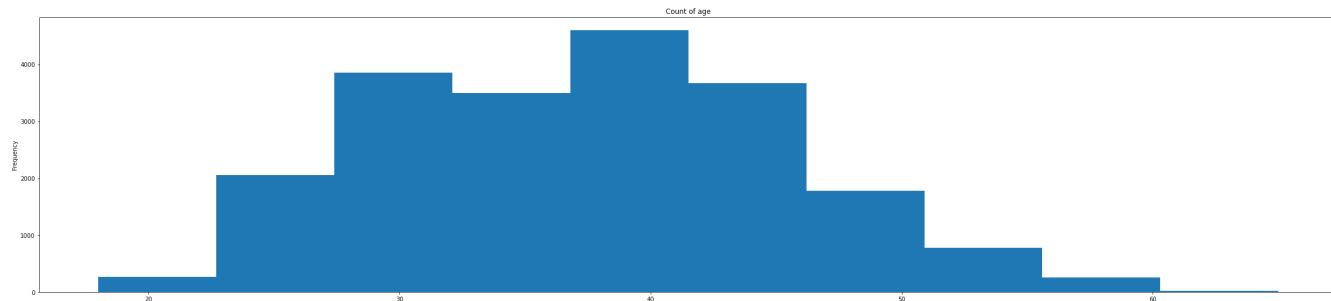
Name: 违法行为_违法小类, dtype: int64

机动车未按规定在限制、禁止的区域通行	8261
不按规定临时停车影响其他车辆和行人通行的	5587
载货汽车、挂车未按照规定安装侧面及后下部防护装置、粘贴车身反光标识的	1242
驾驶机件不符合技术标准的机动车的	936
未将故障车辆移到不妨碍交通的地方停放的	539
不按规定停放影响其他车辆和行人通行的	472
未随车携带驾驶证的	430
机动车号牌不清晰、不完整的	322
未随车携带行驶证的	312

```
Name: 违法行为_违法描述, dtype: int64
```

```
<ipython-input-106-400a247550da>:45: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html#inplace



制作词云图

```
90 import jieba
```

```
df_no_yingyun_value = pd.read_csv("/Users/jiarui/Study/交通事故/data/output/df_no_yingyun_value.csv")
list_word_input = list(df_no_yingyun_value['机动车_交通方式说明'])
```

```
def word_cloud(list_words):
    all_words = []
    for i in list_words:
        words = jieba.lcut(i)
        all_words.append(words)
```

```
print(all_words)

word_cloud(list_word_input)

IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

