Question 3

Data Analysis and Visualization

In [17]:
```
# 1- importing the required libraries
import numpy as np # may be required if we have to deal or initiate a specif
import pandas as pd # for initializaing and dealing with the data recordings
import matplotlib.pyplot as plt # will be required for general plot
import seaborn as sns ## will be required for countplot

```

In [18]:
```
# I have seperated the 2 questions but I will copy and paste the preparation
# last question here
```

In [19]:
```
########################################### Data Preparation ###############
```

In [20]:
```
df = pd.read_csv('data/traffic_violaions.csv')
df = df.drop(['country_name'], axis=1) # remove country_name coulmn
df = df.drop(df.index[[-1,]]) # remove the last raw
df.drop(columns = ['search_type','driver_age_raw','violation_raw'],inplace=T
```

In [21]:
```
mean_value=df['driver_age'].mean()
df['driver_age'].fillna(value=mean_value, inplace=True) # estimate the null
```

In [22]:
```
# remove all rows with Nan values in any columns and check the number of dat
df.dropna(subset=['driver_race','violation','stop_outcome','is_arrested','st
```

In [23]:
```
1 df
```

Out[23]:

	stop_date	stop_time	driver_gender	driver_age	driver_race	violation	search_conducted	s
0	1/2/2005	1:55	M	20.000000	White	Speeding	False	
1	1/18/2005	8:15	M	40.000000	White	Speeding	False	
2	1/23/2005	23:15	M	33.000000	White	Speeding	False	
3	2/20/2005	17:15	M	19.000000	White	Other	False	
4	3/14/2005	10:00	F	21.000000	White	Speeding	False	
...	...	...	...	...	...	...	...	...
52960	10/5/2011	1:55	NaN	34.171483	NaN	NaN	False	
52961	10/5/2011	1:55	NaN	34.171483	NaN	NaN	False	
52962	10/5/2011	6:43	M	25.000000	White	Speeding	False	
52963	10/5/2011	6:49	NaN	34.171483	NaN	NaN	False	
52964	10/5/2011	7:19	F	25.000000	White	Speeding	False	

52965 rows × 11 columns

In [ ]:
```
1
```

In [ ]:
```
1 '''
2 # importing the first 10 raws into an image
3 a = df[0:10]
4 import dataframe_image as dfi
5 dfi.export(a,"headers_10.png")'''
```

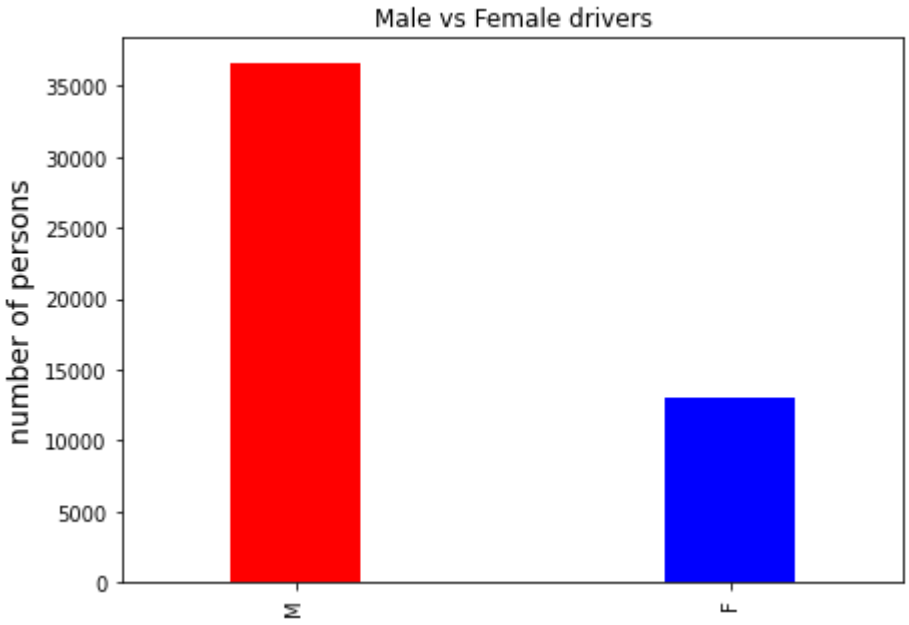
In [26]:
```
1 # now we have clean data points that we can deal with
2 # and data are ready for analysis
```

In [27]:
```
1 ########################################## Data Analysis #################
```

1- Male vs Female drivers who do an accident

In [28]:
```
fig, ax = plt.subplots()
fig.canvas.draw()
df.driver_gender.value_counts().plot(kind="bar", width=0.3, figsize=(7, 5),
plt.title("Male vs Female drivers")
plt.ylabel("number of persons", fontsize=14)
```

Out[28]: Text(27.125, 0.5, 'number of persons')



In [92]:
```
# number of driver genders involved in the dataset
genders = (df.pivot_table(index='driver_gender',columns='driver_gender', agg
genders
```

Out[92]:

driver_gender	driver_gender F	driver_gender M
driver_gender		
F	13016	0
M	0	36564

In [100]:
```
females = genders.values[0][0]
males = genders.values[1][1]
print(f"the percentage of males: {males*100/(females+males)} %")
print(f"the percentage of females: {females*100/(females+males)} %")
```

```
the percentage of males: 73.74747882210569 %
the percentage of females: 26.252521177894312 %
```

In [91]:
```
# the distribution of ages over the genders
pd.crosstab(df['driver_gender'],df['driver_age'], margins=True)
```

Out[91]:

driver_age	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	...	78.0	79.0	80.0	8
driver_gender															
F	2	6	124	364	628	772	745	718	619	578	...	1	3	2	
M	3	20	256	781	1364	1538	1597	1579	1511	1473	...	11	10	8	
All	5	26	380	1145	1992	2310	2342	2297	2130	2051	...	12	13	10	

3 rows × 74 columns

In [63]:
```
genders = df.pivot_table(index=['driver_gender'], aggfunc=np.sum)
num_of_males = int(genders.values[0][0])
num_of_females = int(genders.values[1][0])
total = num_of_males + num_of_females
print(f"Males: {num_of_males}")
print(f"Females: {num_of_females}")
```

```
Males: 420477
Females: 1273782
```

In [10]:
```
male_percent = (df.driver_gender.value_counts().M * 100) / (df.driver_gender
female_percent = 100 - male_percent
```

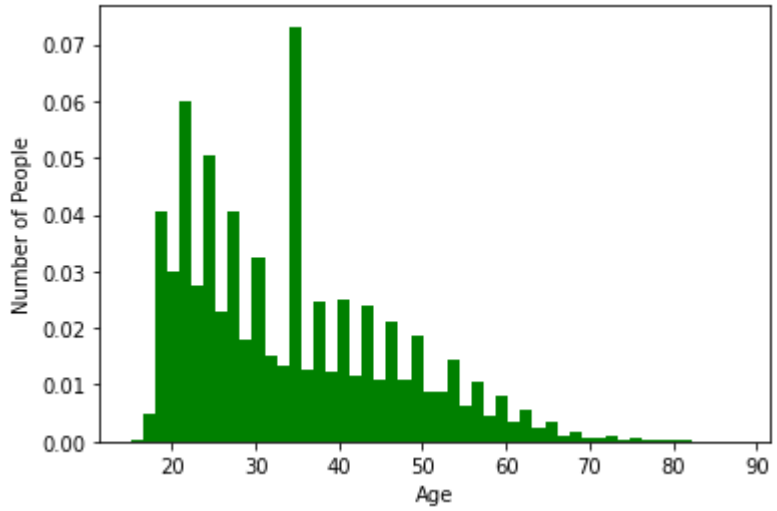
In [11]:
```
print(f"Male Percent: {male_percent} %")
print(f"Female Percent: {female_percent} %")
```

```
Male Percent: 73.74747882210569 %
Female Percent: 26.25252117789431 %
```

2- Variation of people's age vs the number of total people

In [103]:
```
x1 = df.driver_age
x2 = df.loc[df['driver_gender'] == 'M', ['driver_age']]
plt.hist(x1, density=True, bins=50,color='g')  # density=False would make co
plt.ylabel('Number of People')
plt.xlabel('Age')
```

Out[103]: Text(0.5, 0, 'Age')



In [127]:
```
df['driver_age'].value_counts()
```

Out[127]:
```
34.171483    3619
21.000000    2343
20.000000    2310
22.000000    2297
23.000000    2130
             ...
81.000000       5
15.000000       5
84.000000       3
88.000000       2
83.000000       2
Name: driver_age, Length: 73, dtype: int64
```

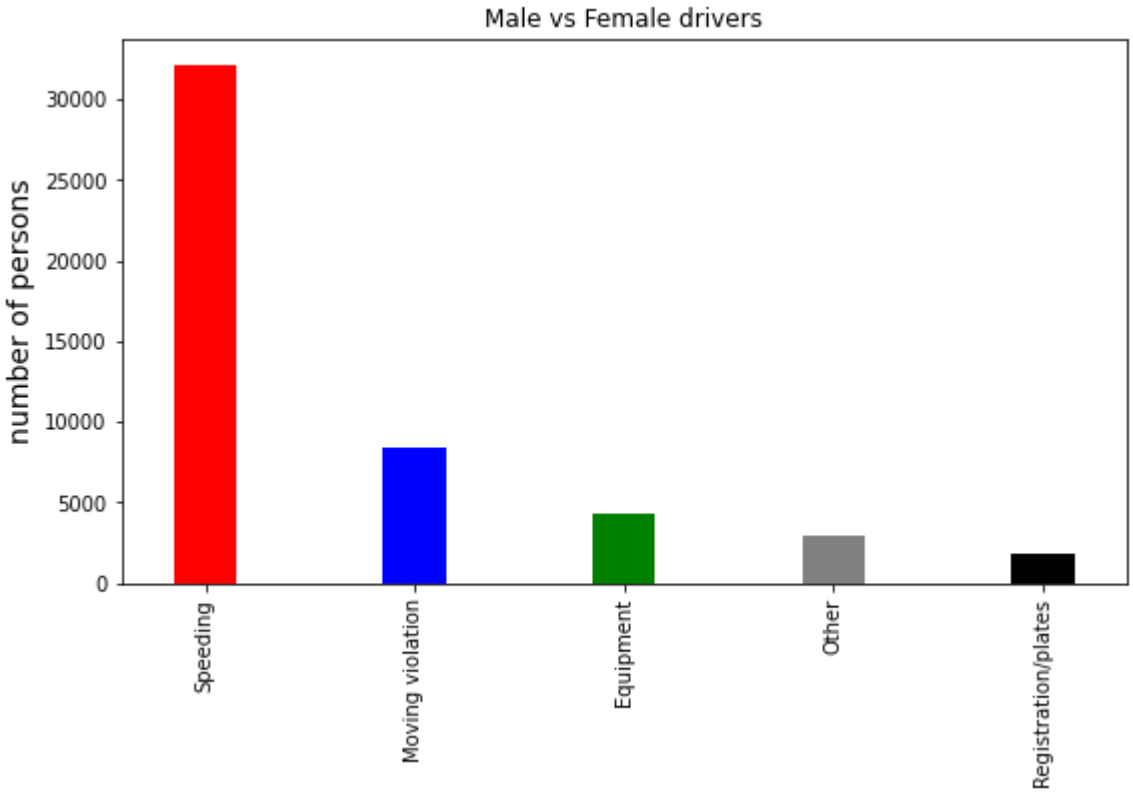
In [128]:
```
# it is clear that the maximum number of drivers who violate the rules were
people_34_ratio = df['driver_age'].value_counts().values[0] * 100 / df['driv
people_34_ratio
```

Out[128]: 6.832814122533748

3- Variations of violation types

In [15]:
```
fig, ax = plt.subplots()
fig.canvas.draw()
df.violation.value_counts().plot(kind="bar", width=0.3, figsize=(9, 5), colo
plt.title("Male vs Female drivers")
plt.ylabel("number of persons", fontsize=14)
```

Out[15]: Text(27.125, 0.5, 'number of persons')



In [155]:
```
pd.crosstab(columns=df['violation'], index='violation', margins=True)
```

Out[155]:

violation	Equipment	Moving violation	Other	Registration/plates	Speeding	All
row_0						
violation	4276	8385	2961	1820	32139	49581
All	4276	8385	2961	1820	32139	49581

In [156]:
```
# from the above crosstable we can coclude the ratio of the speeding violati
speeding_count = pd.crosstab(columns=df['violation'], index='violation', mar
violations_count = pd.crosstab(columns=df['violation'], index='violation', m
print(f"Speeding represents {speeding_count*100/violations_count}% of the to
```

```
Speeding represents 64.82120166999455 % of the total violations
```

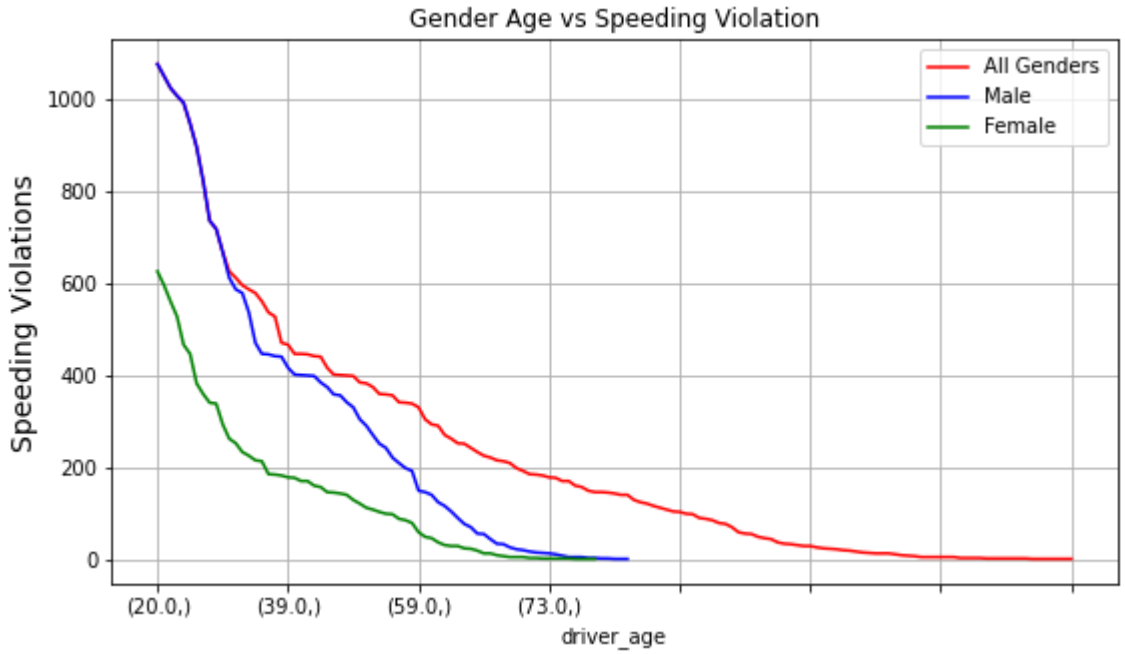
In [ ]:
```

```

4- Age vs Speeding Violation#

In [16]:
```
fig, ax = plt.subplots()
fig.canvas.draw()
x = df.loc[df['violation'] == 'Speeding', ['driver_age','driver_gender']]
male_data = x.loc[x['driver_gender'] == 'M', ['driver_age']]
female_data = x.loc[x['driver_gender'] == 'F', ['driver_age']]
x.value_counts().plot(kind="line",  figsize=(9, 5), color = ["r"])
male_data.value_counts().plot(kind="line",  figsize=(9, 5), color = ["b"])
female_data.value_counts().plot(kind="line",  figsize=(9, 5), color = ["g"])
plt.title("Gender Age vs Speeding Violation")
plt.legend(['All Genders', 'Male','Female'], loc='upper right')
plt.grid()
plt.ylabel("Speeding Violations", fontsize=14)
```

Out[16]: Text(27.125, 0.5, 'Speeding Violations')



In [213]:
```
a = df.pivot_table(index='violation',columns='driver_age', aggfunc='size', f
tot = a.sum()
plt.plot(a.sum())
plt.title("Driver age vs violation")
plt.xlabel("driver age", fontsize=14)
plt.ylabel("stop outcome", fontsize=14)
```

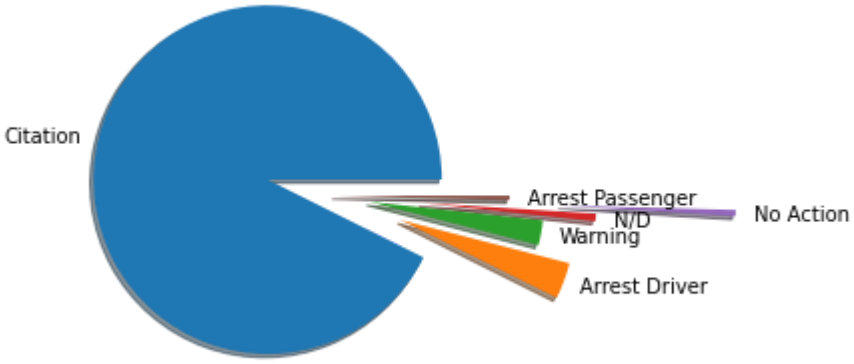
Out[213]: Text(0, 0.5, 'stop outcome')

5- Distribution of Results of Violation

In [17]:
```
import matplotlib.pyplot as plt
import numpy as np

y = df.stop_outcome.value_counts()
mylabels = ["Citation", "Arrest Driver", "Warning", "N/D", "No Action", "Arr
myexplode = [0.4, 0.4, 0.2, 0.5,1.3,0.0]

plt.pie(y, labels = mylabels, explode = myexplode  , shadow = True)
plt.show()
```



In [187]:
```
1 pd.crosstab(columns=df['stop_outcome'], index='stop_outcome', margins=True)
```

Out[187]:

stop_outcome	Arrest Driver	Arrest Passenger	Citation	N/D	No Action	Warning	All
row_0							
stop_outcome	1669	200	45931	348	285	1148	49581
All	1669	200	45931	348	285	1148	49581

In [188]:
```
1 # the ratio of the citation result relative to other stop outcome
2 citation_count = pd.crosstab(columns=df['stop_outcome'], index='stop_outcome
3 total_count = pd.crosstab(columns=df['stop_outcome'], index='stop_outcome',
4 print(f"citation ratio is {(citation_count*100) / (total_count )}%")
```

```
citation ratio is 92.63830902966862%
```

In [189]:
```
1 # the relation between tha age and the stop outcome
2 df.pivot_table(index='stop_outcome',columns='driver_age', aggfunc='size', fi
```

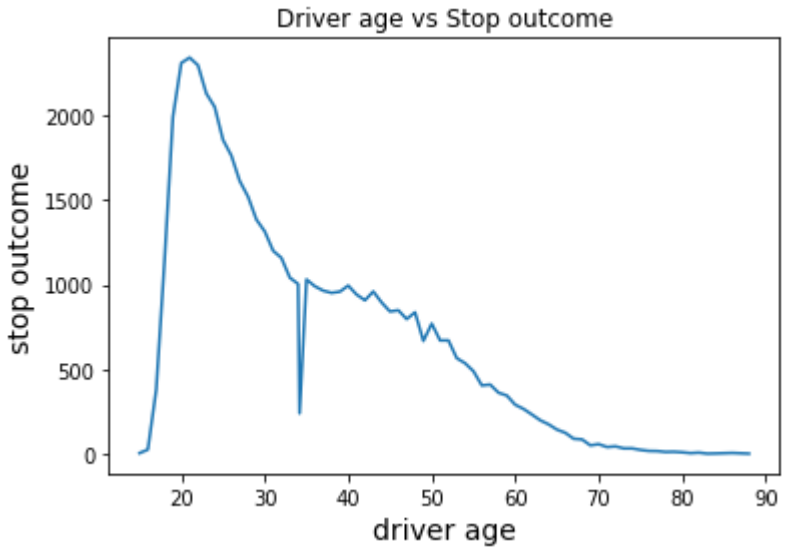
Out[189]:

driver_age	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	...	77.0	78.0	79.0	8
stop_outcome															
Arrest Driver	3	1	6	31	44	77	85	82	76	76	...	0	0	1	
Arrest Passenger	0	0	1	5	12	14	3	11	11	10	...	0	0	0	
Citation	2	21	362	1075	1875	2152	2185	2134	1971	1893	...	15	10	8	
N/D	0	0	1	13	20	15	15	20	20	21	...	0	0	0	
No Action	0	1	1	5	7	10	9	15	8	6	...	0	0	0	
Warning	0	3	9	16	34	42	45	35	44	45	...	2	2	4	

6 rows × 73 columns

In [210]:
```
a = df.pivot_table(index='stop_outcome',columns='driver_age', aggfunc='size'
plt.plot(a.sum())
plt.title("Driver age vs Stop outcome")
plt.xlabel("driver age", fontsize=14)
plt.ylabel("stop outcome", fontsize=14)
```

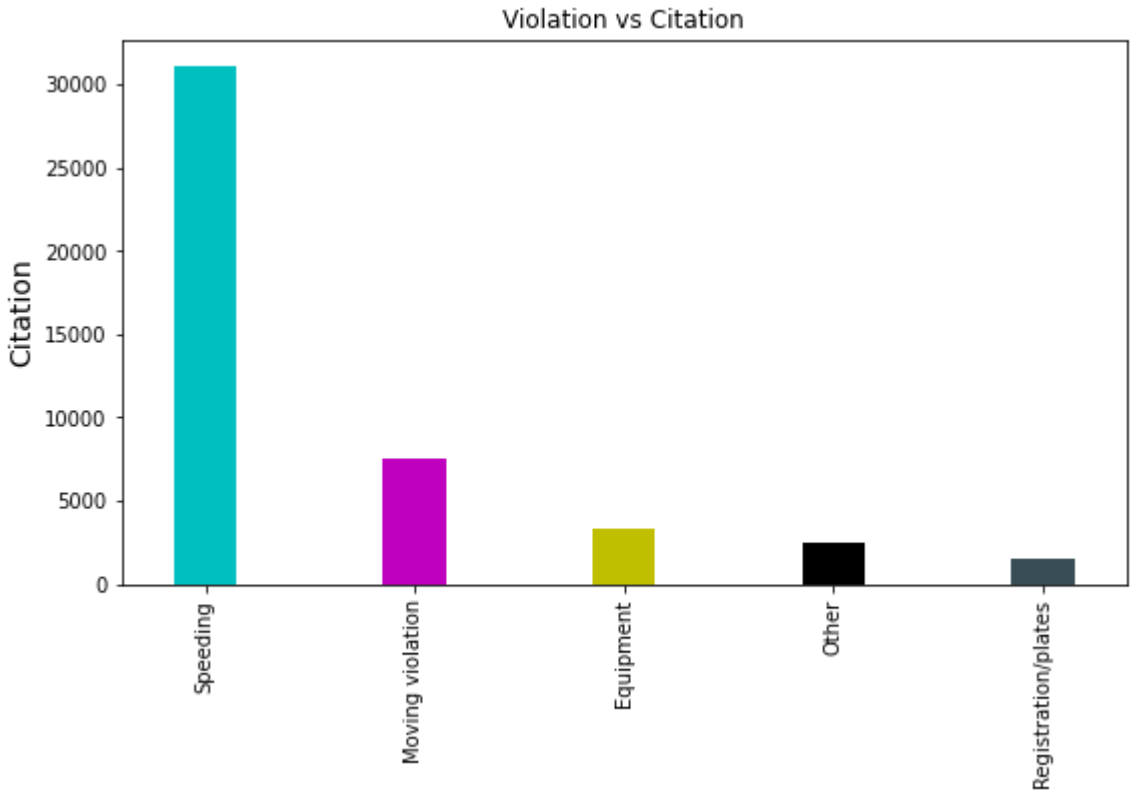
Out[210]: Text(0, 0.5, 'stop outcome')



6- Category of Violation (speed) vs Result of violation (citation)

In [18]:
```
fig, ax = plt.subplots()
fig.canvas.draw()
b = df[df['stop_outcome'] == 'Citation']
b.violation.value_counts().plot(kind="bar", width=0.3, figsize=(9, 5), color
plt.title("Violation vs Citation")
plt.ylabel("Citation", fontsize=14)
```

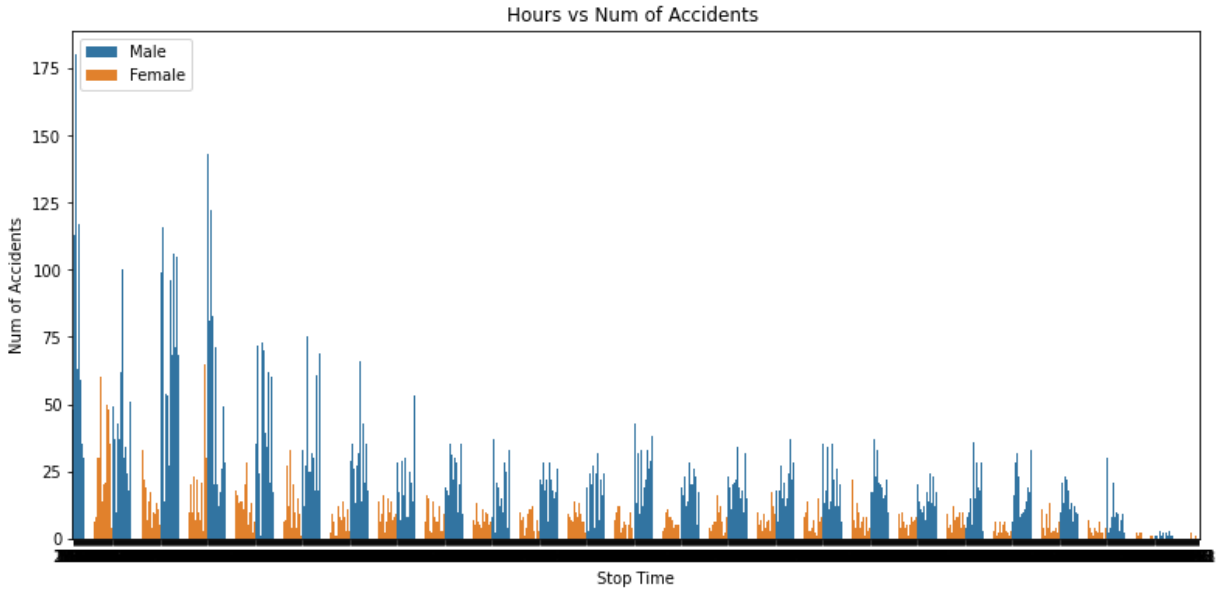
Out[18]: Text(27.125, 0.5, 'Citation')



7- Time where most violations occur

In [222]:
```
plt.figure(figsize = (13,6))
a = df["stop_time"]
sns.countplot(x = a,data = df,hue = 'driver_gender')

plt.title('Hours vs Num of Accidents ')
plt.legend(['Male','Female'])
plt.xlabel("Stop Time")
plt.ylabel("Num of Accidents");
plt.show()
```



In [219]:
```
df['stop_time'].value_counts()
```

Out[219]:
```
11:00    256
10:00    255
10:30    228
9:30     224
9:00     224
        ...
5:31       1
4:37       1
4:41       1
5:21       1
5:23       1
Name: stop_time, Length: 1427, dtype: int64
```

In [220]:
```
print(f"around {(df['stop_time'].value_counts().max()*100/df['stop_time'].va
```

```
around 0.4833380534315137% occurs at 11 o'clock
```

8-Violation vs Stop_outcome

In [214]:
```
df.pivot_table(index='violation',columns='stop_outcome', aggfunc='size', fil
```

Out[214]:

stop_outcome violation	Arrest Driver	Arrest Passenger	Citation	N/D	No Action	Warning
Equipment	281	81	3288	261	25	340
Moving violation	509	47	7506	26	30	267
Other	207	7	2517	8	197	25
Registration/plates	225	23	1512	12	13	35
Speeding	447	42	31108	41	20	481

In [215]:
```
a = df.pivot_table(index='violation',columns='stop_outcome', aggfunc='size',
dfi.export(a,"headers_vilation_vs_stop.png")
```

9-driver_race vs violation

In [227]:
```
df.pivot_table(index='driver_race',columns='violation', aggfunc='size', fill
```

Out[227]:

violation driver_race	Equipment	Moving violation	Other	Registration/plates	Speeding
Asian	92	162	62	25	1034
Black	857	1530	654	428	3125
Hispanic	652	1013	382	358	1694
Other	7	33	19	1	89
White	2668	5647	1844	1008	26197

In [228]:
```
a = df.pivot_table(index='driver_race',columns='violation', aggfunc='size',
dfi.export(a,"driver_race_vs_violation.png")
```

10-Drug vs violation

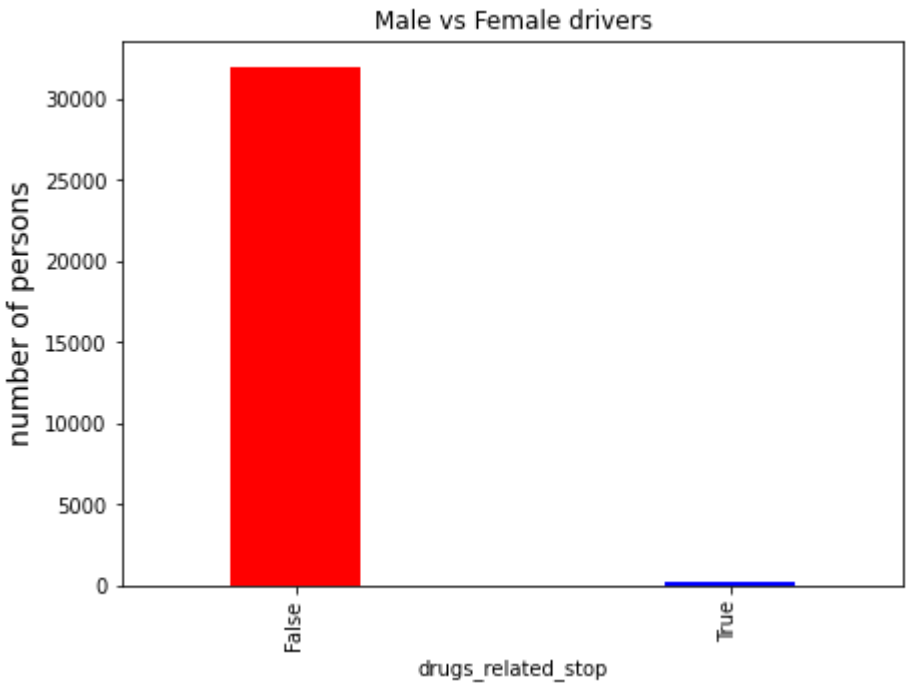
In [229]:
```
# lets draw the relation between the drug and the speeding violation for ins
```

In [240]:
```
a = df.pivot_table(index='drugs_related_stop',columns='violation', aggfunc='
a
```

Out[240]:
```
drugs_related_stop
False    31994
True       145
Name: Speeding, dtype: int64
```

In [242]:
```
fig, ax = plt.subplots()
fig.canvas.draw()
a.plot(kind="bar", width=0.3, figsize=(7, 5), color = ["r","b"])
plt.title("Male vs Female drivers")
plt.ylabel("number of persons", fontsize=14)
```

Out[242]:
```
Text(27.125, 0.5, 'number of persons')
```



In [248]:
```
false_ratio = a[0]*100/(a[0]+a[1])
100-false_ratio
```

Out[248]:
```
0.45116525094123006
```

In [ ]:
```
1
```

In [ ]:
```
1
```