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


# Load the data
Pol_Data=pd.read_csv('Police Data.csv')

# Display first 5 from the dataset
Pol_Data.head(6)
```

	stop_date	stop_time	country_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_
0	1/2/2005	1:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	1/18/2005	8:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	1/23/2005	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	
3	2/20/2005	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	
4	3/14/2005	10:00	NaN	F	1984.0	21.0	White	Speeding	Speeding	
5	3/23/2005	9:45	NaN	M	1982.0	23.0	Black	Equipment/Inspection Violation	Equipment	

```
# Display 4 random samples from the dataset
Pol_Data.sample(4)
```

	stop_date	stop_time	country_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_
32112	1/28/2009	1:45	NaN	M	1961.0	48.0	White	Special Detail/Directed Patrol	Other	
24960	4/7/2008	15:26	NaN	M	1987.0	21.0	White	Speeding	Speeding	
30085	11/1/2008	12:10	NaN	M	1949.0	59.0	White	Speeding	Speeding	
31647	1/10/2009	13:34	NaN	M	1978.0	31.0	White	Speeding	Speeding	

```
# Display the column names
Pol_Data.columns
```

```
Index(['stop_date', 'stop_time', 'country_name', 'driver_gender',
       'driver_age_raw', 'driver_age', 'driver_race', 'violation_raw',
       'violation', 'search_conducted', 'search_type', 'stop_outcome',
       'is_arrested', 'stop_duration', 'drugs_related_stop'],
      dtype='object')
```

```
# Display the shape of the dataset
Pol_Data.shape
```

```
(65535, 15)
```

```
# Display the data types of each column
Pol_Data.dtypes
```

```
stop_date             object
stop_time             object
country_name         float64
driver_gender         object
driver_age_raw       float64
driver_age           float64
driver_race           object
violation_raw         object
violation             object
search_conducted        bool
search_type           object
stop_outcome          object
is_arrested           object
stop_duration         object
drugs_related_stop      bool
dtype: object
```

```
# Count the number of missing values in each column
Pol_Data.isnull().sum()
```

```
stop_date                0
stop_time                0
country_name         65535
driver_gender         4061
driver_age_raw        4054
driver_age            4307
driver_race           4060
violation_raw         4060
violation             4060
search_conducted         0
search_type          63056
stop_outcome          4060
is_arrested           4060
stop_duration         4060
drugs_related_stop       0
dtype: int64
```

```
# Count the number of missing values in each column
Pol_Data.isna().sum()
```

```
stop_date                0
stop_time                0
country_name         65535
driver_gender         4061
driver_age_raw        4054
driver_age            4307
driver_race           4060
violation_raw         4060
violation             4060
search_conducted         0
search_type          63056
stop_outcome          4060
is_arrested           4060
stop_duration         4060
drugs_related_stop       0
dtype: int64
```

```
def duplicatedTotal():
    print(f'Duplicated Data are {Pol_Data.duplicated().sum()}')

duplicatedTotal()

# Display duplicated rows
Pol_Data[Pol_Data.duplicated()]

# Drop duplicates
Pol_Data.drop_duplicates(inplace=True)

# Recheck for duplicates
duplicatedTotal()

# Display the shape after removing duplicates
Pol_Data.shape
```

```
Duplicated Data are 321
Duplicated Data are 0
(65214, 15)
```

```
Pol_Data[Pol_Data.duplicated()]
```

	stop_date	stop_time	country_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_condu

```
def duplicatedTotal():
    print(f'Duplicated Data are now {Pol_Data.duplicated().sum()}')
duplicatedTotal()
```

```
Duplicated Data are now 0
```

Display First 5 Rows After Cleaning:

```
Pol_Data.head(5)
```

	stop_date	stop_time	country_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_
0	1/2/2005	1:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	1/18/2005	8:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	1/23/2005	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	
3	2/20/2005	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	
4	3/14/2005	10:00	NaN	F	1984.0	21.0	White	Speeding	Speeding	

Drop Columns and Check Missing Values:

```
# Drop the 'country_name' column
Pol_Data.drop(columns='country_name', inplace=True)

# Check for missing values
Pol_Data.isnull().sum()
```

```
stop_date                0
stop_time                0
driver_gender         3923
driver_age_raw        3916
driver_age            4169
driver_race           3922
violation_raw         3922
violation             3922
search_conducted         0
search_type          62744
stop_outcome          3922
is_arrested           3922
stop_duration         3922
drugs_related_stop       0
dtype: int64
```

```
Pol_Data
```

	stop_date	stop_time	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conducted
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False
...	...	...	...	...	...	...	...	...	...
65530	12/6/2012	17:54	F	1987.0	25.0	White	Speeding	Speeding	False
65531	12/6/2012	22:22	M	1954.0	58.0	White	Speeding	Speeding	False
65532	12/6/2012	23:20	M	1985.0	27.0	Black	Equipment/Inspection Violation	Equipment	False
65533	12/7/2012	0:23	NaN	NaN	NaN	NaN	NaN	NaN	False
65534	12/7/2012	0:30	F	1985.0	27.0	White	Speeding	Speeding	False

65214 rows × 14 columns

```
Pol_Data.columns
```

```
Index(['stop_date', 'stop_time', 'driver_gender', 'driver_age_raw',
       'driver_age', 'driver_race', 'violation_raw', 'violation',
       'search_conducted', 'search_type', 'stop_outcome', 'is_arrested',
       'stop_duration', 'drugs_related_stop'],
      dtype='object')
```

For Speeding, where men and women stopped often ?

Analyze Speeding Violations:

```
Pol_Data['driver_gender'].value_counts()
```

```
driver_gender
M    45022
F    16269
Name: count, dtype: int64
```

```
# Number of speeding violations
Pol_Data[Pol_Data.violation == "Speeding"].head()

# Count of speeding violations by gender
Pol_Data[Pol_Data.violation == "Speeding"].driver_gender.value_counts()
```

```
driver_gender
M    25441
F    11661
Name: count, dtype: int64
```

```
Pol_Data[Pol_Data.violation=="Speeding"].head()
```

	stop_date	stop_time	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conducted	search_
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	
6	4/1/2005	17:30	M	1969.0	36.0	White	Speeding	Speeding	False	

```
Pol_Data[Pol_Data.violation=="Speeding"].driver_gender.value_counts()
print
```

```
<function print(*args, sep=' ', end='\n', file=None, flush=False)>
```

Who get Searched the most by Gender

Analyze Searches Conducted by Gender:

```
Pol_Data.groupby('driver_gender').search_conducted.sum()
```

```
driver_gender
F     365
M    2105
Name: search_conducted, dtype: int64
```

What is the mean Stop

```
Pol_Data.head(4)
```

	stop_date	stop_time	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conducted	search_
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False	

Analyze Stop Duration:

```
Pol_Data['stop_duration'].value_counts()
```

```
stop_duration
0-15 Min     47264
16-30 Min    11403
30+ Min       2624
2                1
Name: count, dtype: int64
```

```
Pol_Data['stop_duration'].dtype
```

```
dtype('O')
```

```
Pol_Data['stop_duration'] = Pol_Data['stop_duration'].replace({"0-15 Min": 7.5, "16-30 Min": 24, "30+ Min": 45})
```

```
Pol_Data
```

	stop_date	stop_time	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conducted
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False
...	...	...	...	...	...	...	...	...	...
65530	12/6/2012	17:54	F	1987.0	25.0	White	Speeding	Speeding	False
65531	12/6/2012	22:22	M	1954.0	58.0	White	Speeding	Speeding	False
65532	12/6/2012	23:20	M	1985.0	27.0	Black	Equipment/Inspection Violation	Equipment	False
65533	12/7/2012	0:23	NaN	NaN	NaN	NaN	NaN	NaN	False
65534	12/7/2012	0:30	F	1985.0	27.0	White	Speeding	Speeding	False

65214 rows × 14 columns

```
Pol_Data.dtypes
```

```
stop_date              object
stop_time              object
driver_gender          object
driver_age_raw        float64
driver_age            float64
driver_race            object
violation_raw          object
violation              object
search_conducted         bool
search_type            object
stop_outcome           object
is_arrested            object
stop_duration          object
drugs_related_stop       bool
dtype: object
```

```
Pol_Data['stop_duration'] = Pol_Data['stop_duration'].astype(float)
```

```
Pol_Data['stop_duration'].mean()
```

```
np.float64(12.1750636298375)
```

Compare the Age Distrubtion for each violation

Compare Age Distribution for Each Violation:

```
# Describe driver age for each violation
Pol_Data.groupby('violation').driver_age.describe()
```

violation	count	mean	std	min	25%	50%	75%	max
Equipment	6489.0	31.682540	11.383680	16.0	23.0	28.0	39.0	81.0
Moving violation	11844.0	36.742232	13.258955	15.0	25.0	35.0	47.0	86.0
Other	3463.0	40.394456	12.749930	16.0	30.0	41.0	50.0	86.0
Registration/plates	2222.0	32.652115	11.151732	16.0	24.0	30.0	40.0	74.0
Seat belt	3.0	30.333333	10.214369	23.0	24.5	26.0	34.0	42.0
Speeding	37019.0	33.269105	12.618831	15.0	23.0	30.0	42.0	88.0

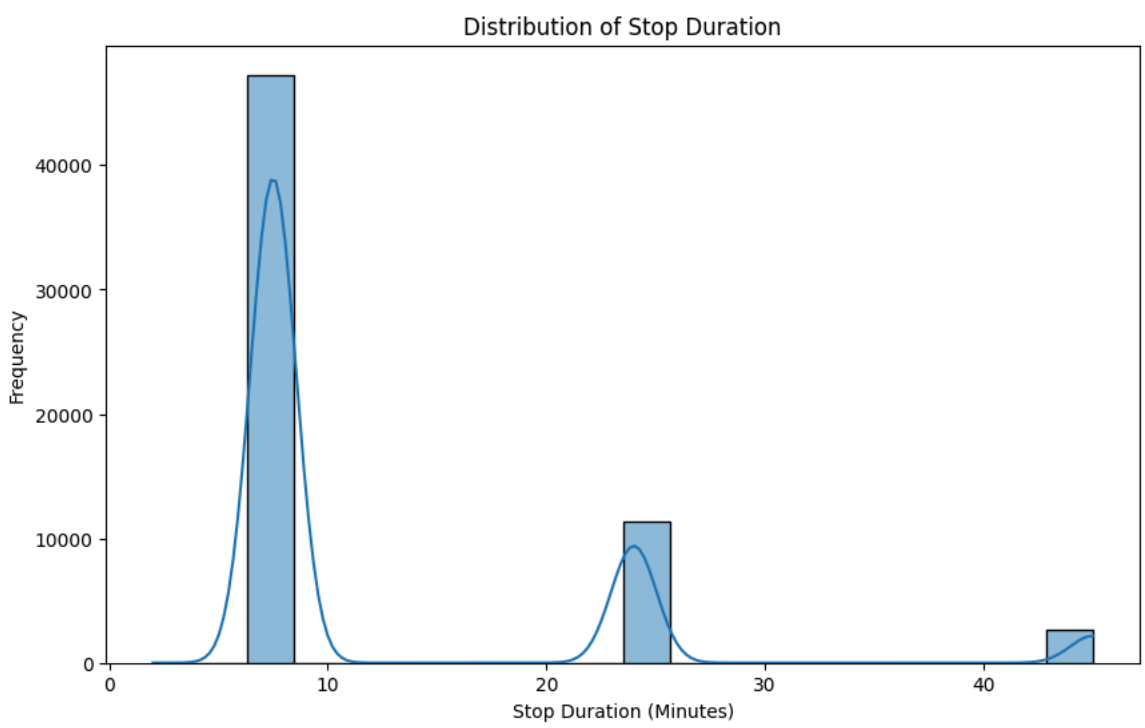
```
# Transpose the description for better readability
Pol_Data.groupby('violation').driver_age.describe().transpose()
```

violation	Equipment	Moving violation	Other	Registration/plates	Seat belt	Speeding
count	6489.00000	11844.000000	3463.000000	2222.000000	3.000000	37019.000000
mean	31.68254	36.742232	40.394456	32.652115	30.333333	33.269105
std	11.38368	13.258955	12.749930	11.151732	10.214369	12.618831
min	16.00000	15.000000	16.000000	16.000000	23.000000	15.000000
25%	23.00000	25.000000	30.000000	24.000000	24.500000	23.000000
50%	28.00000	35.000000	41.000000	30.000000	26.000000	30.000000
75%	39.00000	47.000000	50.000000	40.000000	34.000000	42.000000
max	81.00000	86.000000	86.000000	74.000000	42.000000	88.000000

Distribution of Stop Duration

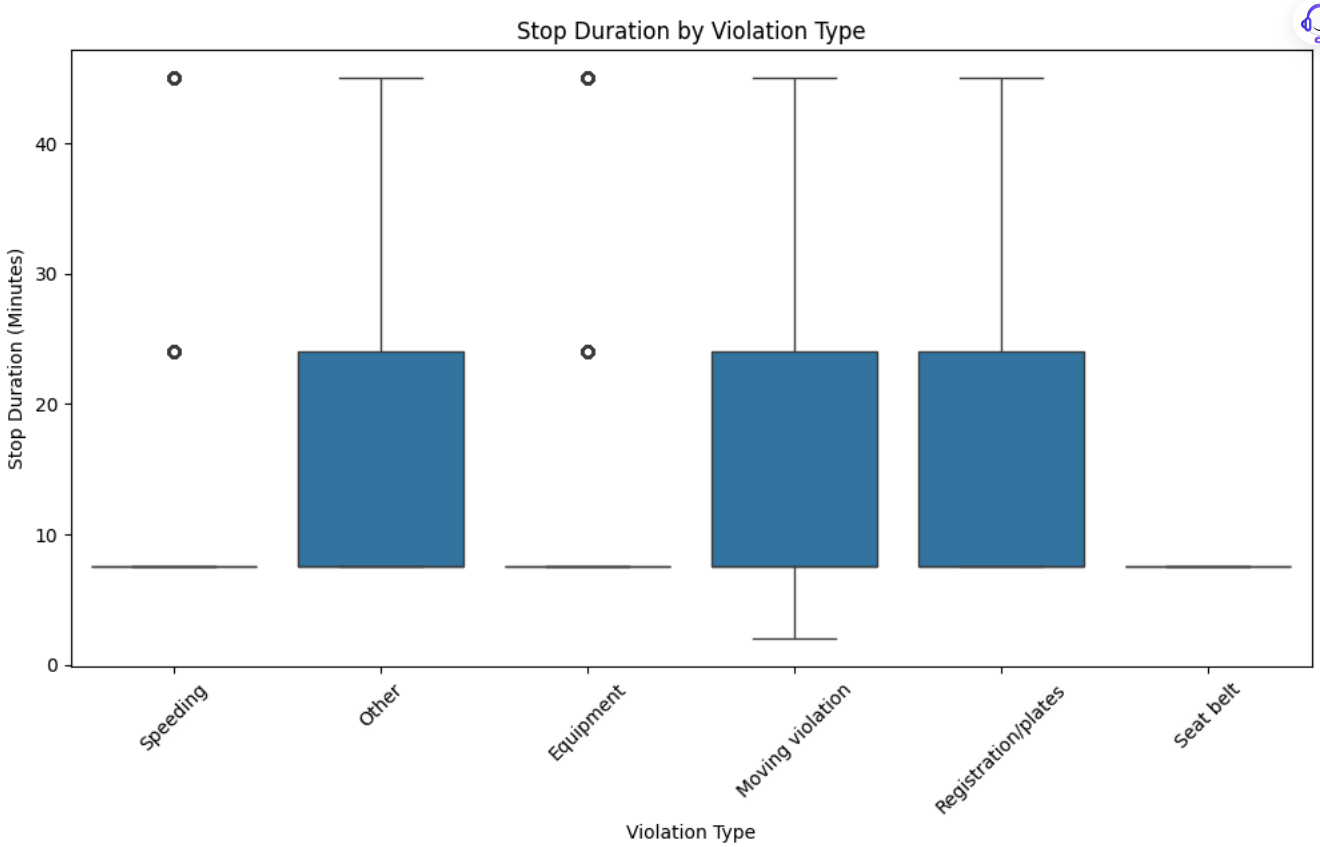
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.histplot(Pol_Data['stop_duration'], bins=20, kde=True)
plt.title('Distribution of Stop Duration')
plt.xlabel('Stop Duration (Minutes)')
plt.ylabel('Frequency')
plt.show()
```



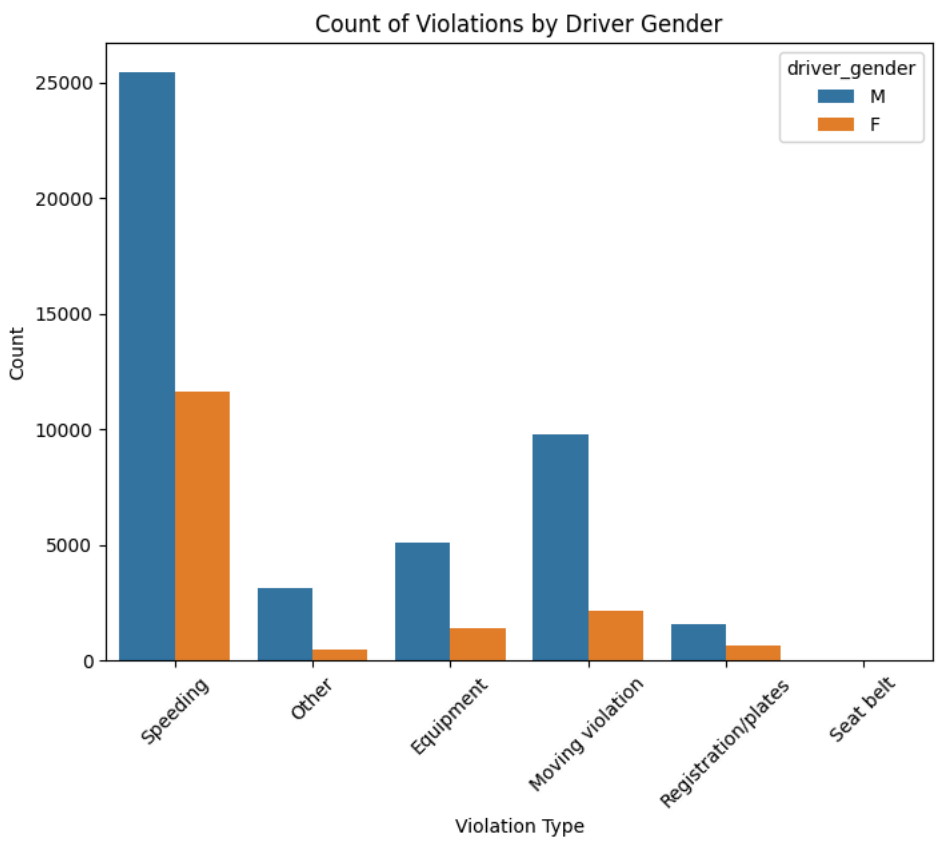
Stop Duration by Violation Type

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='violation', y='stop_duration', data=Pol_Data)
plt.title('Stop Duration by Violation Type')
plt.xlabel('Violation Type')
plt.ylabel('Stop Duration (Minutes)')
plt.xticks(rotation=45)
plt.show()
```



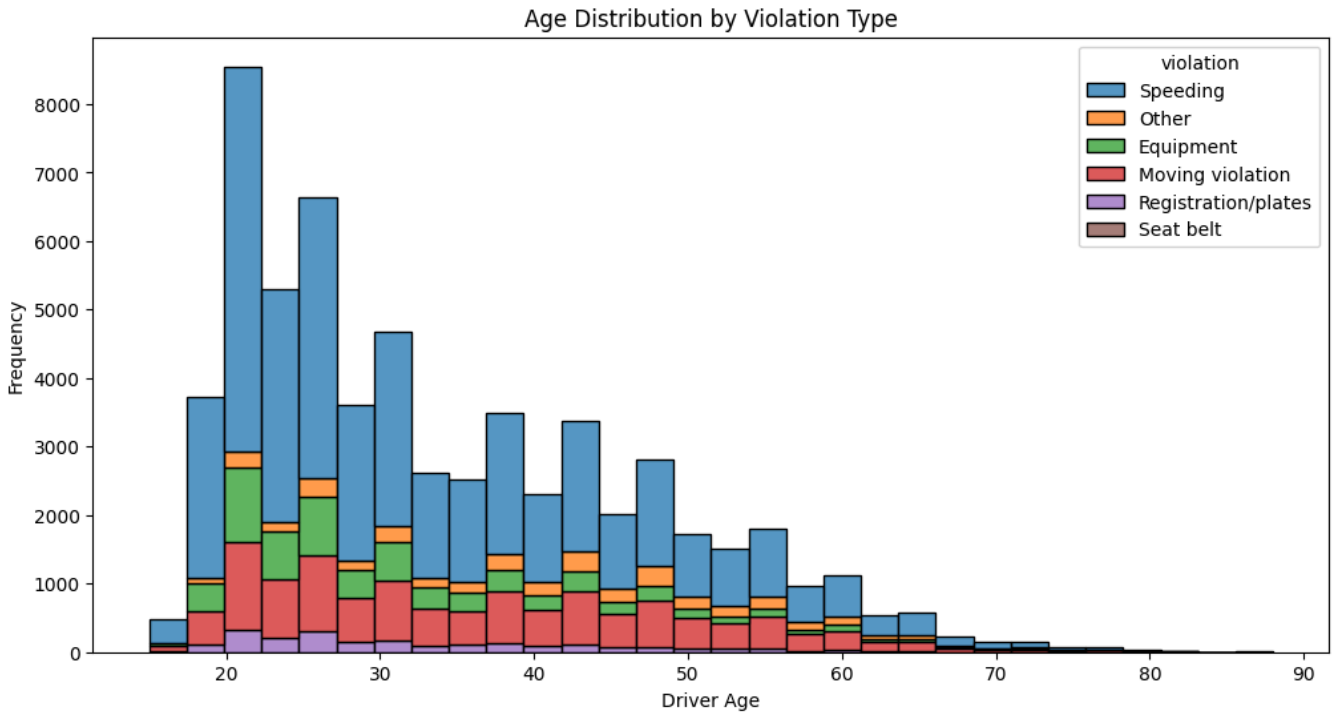
Count of Violations by Gender

```
plt.figure(figsize=(8, 6))
sns.countplot(x='violation', hue='driver_gender', data=Pol_Data)
plt.title('Count of Violations by Driver Gender')
plt.xlabel('Violation Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Age Distribution by Violation Type

```
plt.figure(figsize=(12, 6))
sns.histplot(data=Pol_Data, x='driver_age', hue='violation', multiple='stack', bins=30)
plt.title('Age Distribution by Violation Type')
plt.xlabel('Driver Age')
plt.ylabel('Frequency')
plt.show()
```



Start coding or generate with AI.

Start coding or generate with AI.

This project was crafted by Alpha Yerroh Barrie. I'm excited to showcase the work and its capabilities.

Double-click (or enter) to edit