Police Data set Analysis

In [ ]:
```
# Firstly importing Pandas as pd
```

In [2]:
```
import pandas as pd
```

In [3]:
```
police = pd.read_csv(r"C:\Users\DELL\Desktop\Projects datasets\3. Police Data.csv")
```

In [4]:
```
police
```

Out[4]:

	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
0	1/2/2005	1:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	1/18/2005	8:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	1/23/2005	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2/20/2005	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	3/14/2005	10:00	NaN	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
65530	12/6/2012	17:54	NaN	F	1987.0	25.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
65531	12/6/2012	22:22	NaN	M	1954.0	58.0	White	Speeding	Speeding	False	NaN	Warning	False	0-15 Min	False
65532	12/6/2012	23:20	NaN	M	1985.0	27.0	Black	Equipment/Inspection Violation	Equipment	False	NaN	Citation	False	0-15 Min	False
65533	12/7/2012	0:23	NaN	NaN	NaN	NaN	NaN	NaN	NaN	False	NaN	NaN	NaN	NaN	False
65534	12/7/2012	0:30	NaN	F	1985.0	27.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

65535 rows × 15 columns

In [ ]:
```
# To get the number of column and rows of the data set .shape command is used
```

In [5]:
```
police.shape
```

Out[5]:
```
(65535, 15)
```

Instruction (For data cleaning)

Remove the column that only contains missing values

In [6]:
```
police.head()
```

Out[6]:

	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
0	1/2/2005	1:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	1/18/2005	8:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	1/23/2005	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2/20/2005	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	3/14/2005	10:00	NaN	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

In [ ]:
```
# To find the column that is having no values .isnull().sum() command is used
```

In [7]:
```
police.isnull().sum()
```

Out[7]:
```
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
```

In [ ]:
```
# From the above information the column country_name have no values entered
# So in the below cell executing the code to remove the countr_name column
```

In [12]:
```
police.drop(columns = "country_name" , inplace = True)
```

In [13]:
```
police
```

Out[13]:

	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
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
65530	12/6/2012	17:54	F	1987.0	25.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
65531	12/6/2012	22:22	M	1954.0	58.0	White	Speeding	Speeding	False	NaN	Warning	False	0-15 Min	False
65532	12/6/2012	23:20	M	1985.0	27.0	Black	Equipment/Inspection Violation	Equipment	False	NaN	Citation	False	0-15 Min	False
65533	12/7/2012	0:23	NaN	NaN	NaN	NaN	NaN	NaN	False	NaN	NaN	NaN	NaN	False
65534	12/7/2012	0:30	F	1985.0	27.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

65535 rows × 14 columns

In [14]:
```
# From the data above the required column have been removed
```

Question - based on filtering and value_counts

For speeding, who were stopped more often - Men or women

In [15]:
```
police.head()
```

Out[15]:

	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
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

In [21]:
```
police[police.violation == "Speeding"].driver_gender.value_counts()
```

Out[21]:
```
M    25517
F    11686
Name: driver_gender, dtype: int64
```

In [ ]:
```
# So, clearly from the above the number of men are stopped more often for speeding violation.
```

Does gender affects who gets searched during a stop?

In [22]:
```
police.head()
```

Out[22]:

	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
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

In [23]:
```
police.groupby("driver_gender").search_conducted.sum()
```

Out[23]:
```
driver_gender
F     366
M    2113
Name: search_conducted, dtype: int64
```

In [27]:
```
police.search_conducted.value_counts()
```

Out[27]:
```
False    63056
True      2479
Name: search_conducted, dtype: int64
```

What is the mean for stop_duration

In [28]:
```
police.head()
```

Out[28]:

	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
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

In [31]:
```
police["stop_duration"].mean()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_15888/3376651447.py in <module>
----> 1 police["stop_duration"].mean()

~\anaconda3\lib\site-packages\pandas\core\generic.py in mean(self, axis, skipna, level, numeric_only, **kwargs)
  10749         )
  10750         def mean(self, axis=None, skipna=None, level=None, numeric_only=None, **kwargs):
> 10751             return NDFrame.mean(self, axis, skipna, level, numeric_only, **kwargs)
  10752
  10753         setattr(cls, "mean", mean)

~\anaconda3\lib\site-packages\pandas\core\generic.py in mean(self, axis, skipna, level, numeric_only, **kwargs)
  10367
  10368     def mean(self, axis=None, skipna=None, level=None, numeric_only=None, **kwargs):
> 10369         return self._stat_function(
  10370             "mean", nanops.nanmean, axis, skipna, level, numeric_only, **kwargs
  10371         )

~\anaconda3\lib\site-packages\pandas\core\generic.py in _stat_function(self, name, func, axis, skipna, level, numeric_only, **kwargs)
  10352                 name, axis=axis, level=level, skipna=skipna, numeric_only=numeric_only
  10353             )
> 10354         return self._reduce(
  10355             func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_only
  10356         )

~\anaconda3\lib\site-packages\pandas\core\series.py in _reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4390                 )
   4391             with np.errstate(all="ignore"):
-> 4392                 return op(delegate, skipna=skipna, **kwds)
   4393
   4394     def _reindex_indexer(

~\anaconda3\lib\site-packages\pandas\core\nanops.py in _f(*args, **kwargs)
     91             try:
     92                 with np.errstate(invalid="ignore"):
---> 93                     return f(*args, **kwargs)
     94             except ValueError as e:
     95                 # we want to transform an object array

~\anaconda3\lib\site-packages\pandas\core\nanops.py in f(values, axis, skipna, **kwds)
    153                     result = alt(values, axis=axis, skipna=skipna, **kwds)
    154             else:
--> 155                 result = alt(values, axis=axis, skipna=skipna, **kwds)
    156
    157             return result

~\anaconda3\lib\site-packages\pandas\core\nanops.py in new_func(values, axis, skipna, mask, **kwargs)
    408             mask = isna(values)
    409
--> 410         result = func(values, axis=axis, skipna=skipna, mask=mask, **kwargs)
    411
    412         if datetimelike:

~\anaconda3\lib\site-packages\pandas\core\nanops.py in nanmean(values, axis, skipna, mask)
    663
    664     count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 665     the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    666
    667     if axis is not None and getattr(the_sum, "ndim", False):

~\anaconda3\lib\site-packages\numpy\core\_methods.py in _sum(a, axis, dtype, out, keepdims, initial, where)
     45 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
     46          initial=_NoValue, where=True):
---> 47     return umr_sum(a, axis, dtype, out, keepdims, initial, where)
     48
     49 def _prod(a, axis=None, dtype=None, out=None, keepdims=False,

TypeError: can only concatenate str (not "int") to str
```

In [ ]:
```
# There is an error in the above while executing the code because the format of column is string while it has to be integer.
#so we have to use the map function for changing string to integer.
```

In [38]:
```
police.head()
```

Out[38]:

	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
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

In [39]:
```
police["stop_duration"].value_counts()
```

Out[39]:
```
0-15 Min     47379
16-30 Min    11448
30+ Min       2647
2                1
Name: stop_duration, dtype: int64
```

In [40]:
```
police.stop_duration.value_counts()
```

Out[40]:
```
0-15 Min     47379
16-30 Min    11448
30+ Min       2647
2                1
Name: stop_duration, dtype: int64
```

In [47]:
```
police["stop_duration"] = police["stop_duration"].map({"0-15 Min" : 7.5 , "16-30 Min" : 24 , "30+ Min " : 45})
```

In [48]:
```
police.stop_duration.mean()
```

Out[48]:
```
10.710974552501403
```

In [49]:
```
police["stop_duration"].mean()
```

Out[49]:
```
10.710974552501403
```

Question (groupby, describe())

Compare the age distributions of each violation.

In [50]:
```
police.head()
```

Out[50]:

	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
0	1/2/2005	1:55	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	7.5	False
1	1/18/2005	8:15	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	7.5	False
2	1/23/2005	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	7.5	False
3	2/20/2005	17:15	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	24.0	False
4	3/14/2005	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	7.5	False

In [52]:
```
police.groupby("violation").driver_age.describe()
```

Out[52]:

	count	mean	std	min	25%	50%	75%	max
violation								
Equipment	6507.0	31.682957	11.380671	16.0	23.0	28.0	39.0	81.0
Moving violation	11876.0	36.736443	13.258350	15.0	25.0	35.0	47.0	86.0
Other	3477.0	40.362381	12.754423	16.0	30.0	41.0	50.0	86.0
Registration/plates	2240.0	32.656696	11.150780	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	37120.0	33.262581	12.615781	15.0	23.0	30.0	42.0	88.0

In [53]:
```
police.shape
```

Out[53]:
```
(65535, 14)
```

In [57]:
```
police.describe()
```

Out[57]:

	driver_age_raw	driver_age	stop_duration
count	61481.000000	61228.000000	58827.000000
mean	1967.791106	34.148984	10.710975
std	121.050106	12.760710	6.532339
min	0.000000	15.000000	7.500000
25%	1965.000000	23.000000	7.500000
50%	1978.000000	31.000000	7.500000
75%	1985.000000	43.000000	7.500000
max	8801.000000	88.000000	24.000000

In [ ]: