# Working On Police data analysis Project Using Python

## Police Dataset

Here,

The data from police check point is given.

The dataset is available on CSV format, We are going to anlayze dataset using Pandas Data Frame.

## Import librabry

In [1]:
```
import pandas as pd
```

## Uploading Dataset

In [3]:
```
df = pd,read_csv("C:\Users\Syed Arif\Downloads\3. Police Data.csv")
```

```
  Input In [3]
    df = pd,read_csv("C:\Users\Syed Arif\Downloads\3. Police Data.csv")
                                                                      ^
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in posi
tion 2-3: truncated \UXXXXXXXX escape
```

## Remove Unicode error

We are Written Small r before Quatation

In [5]:
```
df = pd.read_csv(r"C:\Users\Syed Arif\Downloads\3. Police Data.csv")
```

In [6]:
```
df
```

Out[6]:

| r_age_raw | driver_age | driver_race | violation_raw | violation | search_conducted | search_type |
|---|---|---|---|---|---|---|
| 1985.0 | 20.0 | White | Speeding | Speeding | False | NaN |
| 1965.0 | 40.0 | White | Speeding | Speeding | False | NaN |
| 1972.0 | 33.0 | White | Speeding | Speeding | False | NaN |
| 1986.0 | 19.0 | White | Call for Service | Other | False | NaN |
| 1984.0 | 21.0 | White | Speeding | Speeding | False | NaN |
| ... | ... | ... | ... | ... | ... | ... |
| 1987.0 | 25.0 | White | Speeding | Speeding | False | NaN |
| 1954.0 | 58.0 | White | Speeding | Speeding | False | NaN |
| 1985.0 | 27.0 | Black | Equipment/Inspection Violation | Equipment | False | NaN |
| NaN | NaN | NaN | NaN | NaN | False | NaN |
| 1985.0 | 27.0 | White | Speeding | Speeding | False | NaN |

# Data Preprocessing

## .head()

head is used show to the By default = 5 rows in the dataset

In [7]:
```
df.head()
```

Out[7]:

| | stop_date | stop_time | country_name | driver_gender | driver_age_raw | driver_age | driver_race | v |
|---|---|---|---|---|---|---|---|---|
| **0** | 1/2/2005 | 1:55 | NaN | M | 1985.0 | 20.0 | White | |
| **1** | 1/18/2005 | 8:15 | NaN | M | 1965.0 | 40.0 | White | |
| **2** | 1/23/2005 | 23:15 | NaN | M | 1972.0 | 33.0 | White | |
| **3** | 2/20/2005 | 17:15 | NaN | M | 1986.0 | 19.0 | White | |
| **4** | 3/14/2005 | 10:00 | NaN | F | 1984.0 | 21.0 | White | |

## .tail()

tail is used to show rows by Descending order

In [8]:
```
df.tail()
```

Out[8]:

| | stop_date | stop_time | country_name | driver_gender | driver_age_raw | driver_age | driver_rac |
|---|---|---|---|---|---|---|---|
| **65530** | 12/6/2012 | 17:54 | NaN | F | 1987.0 | 25.0 | Whi |
| **65531** | 12/6/2012 | 22:22 | NaN | M | 1954.0 | 58.0 | Whi |
| **65532** | 12/6/2012 | 23:20 | NaN | M | 1985.0 | 27.0 | Blac |
| **65533** | 12/7/2012 | 0:23 | NaN | NaN | NaN | NaN | Na |
| **65534** | 12/7/2012 | 0:30 | NaN | F | 1985.0 | 27.0 | Whi |

# .shape

It shoe the total no of rows & Column in the dataset

In [9]:
```
df.shape
```

Out[9]:
```
(65535, 15)
```

# .Columns

It show the no of each Column

In [10]:
```
df.columns
```

Out[10]:
```
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')
```

# .dtypes

This Attribute show the data type of each column

In [11]:
```
df.dtypes
```

Out[11]:
```
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
```

# .unique()

In a column, It show the unique value of specific column.

In [13]:
```
df['country_name'].unique
```

Out[13]:
```
<bound method Series.unique of 0       NaN
1       NaN
2       NaN
3       NaN
4       NaN
         ..
65530   NaN
65531   NaN
65532   NaN
65533   NaN
65534   NaN
Name: country_name, Length: 65535, dtype: float64>
```

In [14]:
```
df['country_name'].unique()
```

Out[14]:
```
array([nan])
```

# nuique()

It will show the total no of unque value from whole data frame.

In [15]:
```
df.nunique()
```

Out[15]:
```
stop_date             2651
stop_time             1432
country_name             0
driver_gender            2
driver_age_raw          93
driver_age              73
driver_race              5
violation_raw           12
violation                6
search_conducted         2
search_type             23
stop_outcome             6
is_arrested              2
stop_duration            4
drugs_related_stop       2
dtype: int64
```

# .describe()

It show the Count, mean , median etc

In [16]:
```
df.describe()
```

Out[16]:

| | country_name | driver_age_raw | driver_age |
|---|---|---|---|
| **count** | 0.0 | 61481.000000 | 61228.000000 |
| **mean** | NaN | 1967.791106 | 34.148984 |
| **std** | NaN | 121.050106 | 12.760710 |
| **min** | NaN | 0.000000 | 15.000000 |
| **25%** | NaN | 1965.000000 | 23.000000 |
| **50%** | NaN | 1978.000000 | 31.000000 |
| **75%** | NaN | 1985.000000 | 43.000000 |
| **max** | NaN | 8801.000000 | 88.000000 |

# .value_counts

It Shows all the unique values with their count

In [17]:
```
df["driver_race"].value_counts()
```

Out[17]:
```
White       45747
Black        8267
Hispanic     5611
Asian        1639
Other         211
Name: driver_race, dtype: int64
```

# Q : 1 ) Remove the column that only contains misssing values

# .isnull()

Find out all the Null values in the data frame

In [19]:
```
df.isnull().sum()
```

Out[19]:
```
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
```

There are many null values in this data frame Heighest Null Values is (country_name == 65535 ) Columns That's Why we Dop the column Country_name

In [20]:
```
df.drop(columns = country_name , inplace  = True)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [20], in <cell line: 1>()
----> 1 df.drop(columns = country_name , inplace  = True)

NameError: name 'country_name' is not defined
```

# Why we are facing this error.....?

Because Country_name is always written in string form == "country_name"

In [22]:
```
df.drop(columns = "country_name" , inplace  = True)
```

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

Out[23]:

| r_age_raw | driver_age | driver_race | violation_raw | violation | search_conducted | search_type |
|---|---|---|---|---|---|---|
| 1985.0 | 20.0 | White | Speeding | Speeding | False | NaN |
| 1965.0 | 40.0 | White | Speeding | Speeding | False | NaN |
| 1972.0 | 33.0 | White | Speeding | Speeding | False | NaN |
| 1986.0 | 19.0 | White | Call for Service | Other | False | NaN |
| 1984.0 | 21.0 | White | Speeding | Speeding | False | NaN |
| ... | ... | ... | ... | ... | ... | ... |
| 1987.0 | 25.0 | White | Speeding | Speeding | False | NaN |
| 1954.0 | 58.0 | White | Speeding | Speeding | False | NaN |
| 1985.0 | 27.0 | Black | Equipment/Inspection Violation | Equipment | False | NaN |
| NaN | NaN | NaN | NaN | NaN | False | NaN |
| 1985.0 | 27.0 | White | Speeding | Speeding | False | NaN |

# Q : 2 ) For Spreeding , were men or women stopped more often ?

In [29]:
```
df[df.violation == "Speeding"].driver_gender.value_counts()
```

Out[29]:
```
M    25517
F    11686
Name: driver_gender, dtype: int64
```

# Q : 3 ) Who gender affect who gets searched during a stop ?

In [31]:
```
df.groupby("driver_gender").search_conducted.sum()
```

Out[31]:
```
driver_gender
F     366
M    2113
Name: search_conducted, dtype: int64
```

# Q : 4 ) What is mean stop_duration ?

In [36]:
```
df['stop_duration'].mean()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [36], in <cell line: 1>()
----> 1 df['stop_duration'].mean()

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:11117, in NDFrame._
add_numeric_operations.<locals>.mean(self, axis, skipna, level, numeric_only,
**kwargs)
  11099 @doc(
  11100     _num_doc,
  11101     desc="Return the mean of the values over the requested axis.",
   (...)
  11115     **kwargs,
  11116 ):
> 11117     return NDFrame.mean(self, axis, skipna, level, numeric_only, **kw
args)

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:10687, in NDFrame.m
ean(self, axis, skipna, level, numeric_only, **kwargs)
  10679 def mean(
  10680     self,
  10681     axis: Axis | None | lib.NoDefault = lib.no_default,
   (...)
  10685     **kwargs,
  10686 ) -> Series | float:
> 10687     return self._stat_function(
  10688         "mean", nanops.nanmean, axis, skipna, level, numeric_only, **
kwargs
  10689     )

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:10639, in NDFrame._
stat_function(self, name, func, axis, skipna, level, numeric_only, **kwargs)
  10629     warnings.warn(
  10630         "Using the level keyword in DataFrame and Series aggregations
is "
  10631         "deprecated and will be removed in a future version. Use grou
pby "
   (...)
  10634         stacklevel=find_stack_level(),
  10635     )
  10636     return self._agg_by_level(
  10637         name, axis=axis, level=level, skipna=skipna, numeric_only=num
eric_only
  10638     )
> 10639 return self._reduce(
  10640     func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_o
nly
  10641 )

File ~\anaconda3\lib\site-packages\pandas\core\series.py:4471, in Series._red
uce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4467     raise NotImplementedError(
   4468         f"Series.{name} does not implement {kwd_name}."
   4469     )
   4470 with np.errstate(all="ignore"):
-> 4471     return op(delegate, skipna=skipna, **kwds)
```

```
File ~\anaconda3\lib\site-packages\pandas\core\nanops.py:93, in disallow.__ca
ll__.<locals>._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
     96     # ValueError message to the more typical TypeError
     97     # e.g. this is normally a disallowed function on
     98     # object arrays that contain strings
     99     if is_object_dtype(args[0]):

File ~\anaconda3\lib\site-packages\pandas\core\nanops.py:155, in bottleneck_s
witch.__call__.<locals>.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)
    157 return result

File ~\anaconda3\lib\site-packages\pandas\core\nanops.py:410, in _datetimelik
e_compat.<locals>.new_func(values, axis, skipna, mask, **kwargs)
    407 if datetimelike and mask is None:
    408     mask = isna(values)
--> 410 result = func(values, axis=axis, skipna=skipna, mask=mask, **kwargs)
    412 if datetimelike:
    413     result = _wrap_results(result, orig_values.dtype, fill_value=iNa
T)

File ~\anaconda3\lib\site-packages\pandas\core\nanops.py:698, in nanmean(valu
es, axis, skipna, mask)
    695     dtype_count = dtype
    697 count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 698 the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    700 if axis is not None and getattr(the_sum, "ndim", False):
    701     count = cast(np.ndarray, count)

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

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

# Reason of this type of error ?

Beacause the column Stop_duration is String format , Before we take an output it will convert string to int

In [37]:
```
df.stop_duration.value_counts()
```

Out[37]:
```
0-15 Min     47379
16-30 Min    11448
30+ Min       2647
2                1
Name: stop_duration, dtype: int64
```

In [39]:
```
df['stop_duration'] = df["stop_duration"].map({"0-15 Min" : 8 , "16-30 Min" :
```

In [40]:
```
df
```

Out[40]:

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

65535 rows × 14 columns

In [41]:
```
df["stop_duration"].mean()
```

Out[41]:
```
11.308276811668112
```

# Compare the age distribuation of each violation ?

In [44]:
```
df.groupby("violation").driver_age.describe()
```

Out[44]:

| violation | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **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 [ ]:
```

```