Task2(Python Programming)

Reading csv file using pandas

```
import pandas as pd
import os

# Function to get the full file path
def get_file_path(filename):
    current_dir = os.getcwd()  # Get the current working directory
    full_path = os.path.join(current_dir, filename)  # Join the directory with the filename
    return full_path

# Load the CSV file into a DataFrame
df = pd.read_csv(get_file_path('police.csv'))

# Print the first 5 rows of the DataFrame
print(df.head(5))
```

```
    stop_date stop_time  county_name driver_gender  driver_age_raw  \
0  2005-01-02     01:55          NaN             M          1985.0
1  2005-01-18     08:15          NaN             M          1965.0
2  2005-01-23     23:15          NaN             M          1972.0
3  2005-02-20     17:15          NaN             M          1986.0
4  2005-03-14     10:00          NaN             F          1984.0

   driver_age driver_race     violation_raw violation  search_conducted  \
0        20.0       White          Speeding  Speeding             False
1        40.0       White          Speeding  Speeding             False
2        33.0       White          Speeding  Speeding             False
3        19.0       White  Call for Service     Other             False
4        21.0       White          Speeding  Speeding             False

  search_type   stop_outcome is_arrested stop_duration  drugs_related_stop
0         NaN       Citation       False      0-15 Min               False
1         NaN       Citation       False      0-15 Min               False
2         NaN       Citation       False      0-15 Min               False
3         NaN  Arrest Driver        True     16-30 Min               False
4         NaN       Citation       False      0-15 Min               False
```

Using pandas “head()” function to display the top 5 rows from our data set.

Last 5 rows of the data set

```
df.tail(5)
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	dri
91736	2015-12-31	20:27	NaN	M	1986.0	29.0	
91737	2015-12-31	20:35	NaN	F	1982.0	33.0	
91738	2015-12-31	20:45	NaN	M	1992.0	23.0	
91739	2015-12-31	21:42	NaN	M	1993.0	22.0	
91740	2015-12-31	22:46	NaN	M	1959.0	56.0	

Dimension of the Data set and information about dataset

```
df.shape
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91741 entries, 0 to 91740
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   stop_date           91741 non-null  object 
 1   stop_time           91741 non-null  object 
 2   county_name         0 non-null      float64
 3   driver_gender       86406 non-null  object 
 4   driver_age_raw      86414 non-null  float64
 5   driver_age          86120 non-null  float64
 6   driver_race         86408 non-null  object 
 7   violation_raw       86408 non-null  object 
 8   violation           86408 non-null  object 
 9   search_conducted    91741 non-null  bool   
 10  search_type         3196 non-null   object 
 11  stop_outcome        86408 non-null  object 
 12  is_arrested         86408 non-null  object 
 13  stop_duration       86408 non-null  object 
 14  drugs_related_stop  91741 non-null  bool   
dtypes: bool(2), float64(3), object(10)
memory usage: 9.3+ MB
```

```
#missing values in columns
df.isnull().sum()
```

```
stop_date                0
stop_time                0
county_name          91741
driver_gender         5335
driver_age_raw        5327
driver_age            5621
driver_race           5333
violation_raw         5333
violation             5333
search_conducted         0
search_type          88545
stop_outcome          5333
is_arrested           5333
stop_duration         5333
drugs_related_stop       0
dtype: int64
```

```
df.describe()
```

	county_name	driver_age_raw	driver_age
count	0.0	86414.000000	86120.000000
mean	NaN	1970.491228	34.011333
std	NaN	110.914909	12.738564
min	NaN	0.000000	15.000000
25%	NaN	1967.000000	23.000000
50%	NaN	1980.000000	31.000000
75%	NaN	1987.000000	43.000000
max	NaN	8801.000000	99.000000

```
df.sample(5)
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conduc
71435	2013-09-04	22:30	NaN	M	1964.0	49.0	White	Speeding	Speeding	F
88231	2015-08-12	23:11	NaN	F	1982.0	33.0	Hispanic	Speeding	Speeding	F
40855	2010-03-05	08:56	NaN	M	1966.0	44.0	White	Special Detail/Directed Patrol	Other	F
33630	2009-03-28	11:35	NaN	M	1985.0	24.0	Hispanic	Other Traffic Violation	Moving violation	F
20527	2007-10-01	07:22	NaN	M	1958.0	49.0	White	Speeding	Speeding	F

loc() and iloc() methods are used in slicing data from the pandas DataFrame

```
df.loc[:5,['driver_age','violation']]
```

	driver_age	violation
0	20.0	Speeding
1	40.0	Speeding
2	33.0	Speeding
3	19.0	Other
4	21.0	Speeding
5	23.0	Equipment

```
df.loc[(df.driver_age <16)& (df.violation == 'Speeding')]
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conduc
17771	2007-06-11	12:30	NaN	M	1992.0	15.0	White	Speeding	Speeding	F

```
df.iloc[:5,:5]
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw
0	2005-01-02	01:55	NaN	M	1985.0
1	2005-01-18	08:15	NaN	M	1965.0
2	2005-01-23	23:15	NaN	M	1972.0
3	2005-02-20	17:15	NaN	M	1986.0
4	2005-03-14	10:00	NaN	F	1984.0

We can sort our DataFrame by index or values with Pandas “sort_index()” and “sort_values()” functions

```
df[['stop_date','driver_age','driver_race','violation','stop_outcome']].sort_values(by='driver_age')
```

	stop_date	driver_age	driver_race	violation	stop_outcome
18357	2007-07-04	15.0	White	Moving violation	Arrest Driver
17771	2007-06-11	15.0	White	Speeding	Citation
45988	2010-11-10	15.0	White	Moving violation	Citation
10500	2006-09-30	15.0	Black	Moving violation	Arrest Driver
25294	2008-04-21	15.0	Hispanic	Moving violation	Arrest Driver
...	...	...	...	...	...
91637	2015-12-27	NaN	NaN	NaN	NaN
91660	2015-12-28	NaN	NaN	NaN	NaN
91674	2015-12-28	NaN	NaN	NaN	NaN
91710	2015-12-30	NaN	NaN	NaN	NaN
91713	2015-12-30	NaN	NaN	NaN	NaN

91741 rows × 5 columns

We can use the Pandas query() function to filter our data frame as per our conditions

```
df.query('45<driver_age <50').head()
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_cond
19	2005-07-24	20:10	NaN	F	1958.0	47.0	White	Speeding	Speeding	
30	2005-09-26	12:09	NaN	M	1959.0	46.0	Black	Other Traffic Violation	Moving violation	
46	2005-10-01	08:40	NaN	M	1959.0	46.0	White	Equipment/Inspection Violation	Equipment	
48	2005-10-01	09:20	NaN	M	1957.0	48.0	White	Speeding	Speeding	
57	2005-10-01	18:10	NaN	M	1958.0	47.0	White	Speeding	Speeding	

Filtering based on the condition

```
# Filter rows where 'driver_gender' is 'M' (Male)
male_drivers = df[df['driver_gender'] == 'M']
print("\nMale drivers:")
print(male_drivers.head())  # Print the first 5 rows of male drivers

# Filter rows where 'violation' is 'Speeding' and 'is_arrested' is True
speeding_arrests = df[(df['violation'] == 'Speeding') & (df['is_arrested'] == True)]
print("\nSpeeding violations resulting in arrest:")
print(speeding_arrests.head())  # Print the first 5 rows of speeding violations that resulted in arrest
```

```
Male drivers:
    stop_date stop_time  county_name driver_gender  driver_age_raw  \
0  2005-01-02     01:55          NaN             M          1985.0
1  2005-01-18     08:15          NaN             M          1965.0
2  2005-01-23     23:15          NaN             M          1972.0
3  2005-02-20     17:15          NaN             M          1986.0
5  2005-03-23     09:45          NaN             M          1982.0

   driver_age driver_race                  violation_raw  violation  \
0        20.0       White                       Speeding   Speeding
1        40.0       White                       Speeding   Speeding
2        33.0       White                       Speeding   Speeding
3        19.0       White                Call for Service      Other
5        23.0       Black  Equipment/Inspection Violation  Equipment

   search_conducted search_type   stop_outcome is_arrested stop_duration  \
0             False         NaN       Citation       False      0-15 Min
```

```
1             False         NaN       Citation       False      0-15 Min
2             False         NaN       Citation       False      0-15 Min
3             False         NaN  Arrest Driver        True     16-30 Min
5             False         NaN       Citation       False      0-15 Min

   drugs_related_stop
0               False
1               False
2               False
3               False
5               False

Speeding violations resulting in arrest:
      stop_date stop_time  county_name driver_gender  driver_age_raw  \
31   2005-09-28     06:20          NaN             M          1982.0
80   2005-10-02     09:30          NaN             M          1975.0
103  2005-10-03     13:26          NaN             M          1975.0
104  2005-10-03     13:26          NaN             M          1975.0
131  2005-10-04     15:00          NaN             M          1981.0

     driver_age driver_race violation_raw violation  search_conducted  \
31         23.0       White      Speeding  Speeding             False
80         30.0       White      Speeding  Speeding              True
103        30.0       Black      Speeding  Speeding             False
104        30.0       Black      Speeding  Speeding             False
131        24.0       Black      Speeding  Speeding             False

             search_type   stop_outcome is_arrested stop_duration  \
31                   NaN  Arrest Driver        True     16-30 Min
80   Incident to Arrest  Arrest Driver        True       30+ Min
103                  NaN  Arrest Driver        True       30+ Min
104                  NaN  Arrest Driver        True       30+ Min
131                  NaN  Arrest Driver        True       30+ Min

     drugs_related_stop
31                False
80                False
103               False
104               False
131               False
```

missing values

```
# Example: Count missing values in each column
missing_values = df.isnull().sum()
print("\nMissing values in each column:")
print(missing_values)

# Example: Fill missing values in 'county_name' with a default value
df['county_name'].fillna('Unknown', inplace=True)
```

```
Missing values in each column:
stop_date                0
stop_time                0
county_name          91741
driver_gender         5335
driver_age_raw        5327
driver_age            5621
driver_race           5333
violation_raw         5333
violation             5333
search_conducted         0
search_type          88545
stop_outcome          5333
is_arrested           5333
stop_duration         5333
drugs_related_stop       0
dtype: int64
```

calculating summary statics

```
# Calculate mean age of drivers
mean_age = df['driver_age'].mean()
print(f"\nMean age of drivers: {mean_age}")

# Example: Calculate count of each unique value in 'stop_outcome'
outcome_counts = df['stop_outcome'].value_counts()
print("\nStop outcomes count:")
```