Loading data in csv

In [50]:
```
import pandas as pd
import re
import warnings
warnings.filterwarnings('ignore')
data=pd.read_csv("combined_data.csv")
data.head()
```

Out[50]:

	Unnamed: 0	DATE_REPORTED	DATE_OCCURRED	TIME_OCCURRED	PATROL_DIVISION	AREA_
0	0	2008-04-10	2008-04-10	19:10:00	73.0	BRO
1	1	2007-06-03	2007-06-03	15:23:00	28.0	MANH
2	2	2010-02-16	2010-02-16	20:50:00	102.0	QU
3	3	2009-11-10	2009-11-10	16:35:00	79.0	BRO
4	4	2006-04-25	2006-04-11	09:30:00	123.0	S IS

Drop column

In [51]:
```
data.drop(data.columns[0], axis=1, inplace=True)
```

In [52]:
```
data
```

Out[52]:

	DATE_REPORTED	DATE_OCCURRED	TIME_OCCURRED	PATROL_DIVISION	AREA_NAME	C
0	2008-04-10	2008-04-10	19:10:00	73.0	BROOKLYN	
1	2007-06-03	2007-06-03	15:23:00	28.0	MANHATTAN	
2	2010-02-16	2010-02-16	20:50:00	102.0	QUEENS	
3	2009-11-10	2009-11-10	16:35:00	79.0	BROOKLYN	
4	2006-04-25	2006-04-11	09:30:00	123.0	STATEN ISLAND	
...	...	...	...	...	...	
197	2010-04-01	2010-03-29	2320	1.0	Central	
198	2010-03-31	2010-03-31	1515	1.0	Central	
199	2010-04-03	2010-04-02	2230	1.0	Central	
200	2010-04-03	2010-04-03	1	1.0	Central	
201	2010-04-06	2010-04-05	1940	1.0	Central	

202 rows × 13 columns

Cleaning attributes

TIME_OCCURRED: is in a "hh:mm:ss" format for the new york dataset whereas military time format for LA dataset. Converting the new york dataset to military time

In [53]:
```
for index,val in enumerate(data["TIME_OCCURRED"]):
    data["TIME_OCCURRED"][index]=re.sub('^(?:(?:([01]?\d|2[0-3]):)?([0-5]?\d):)?([0-5]?\d)$', "".join(val.split(":")[:-1]), val)
```

In [54]:
```
data["TIME_OCCURRED"]
```

Out[54]:
```
0      1910
1      1523
2      2050
3      1635
4      0930
       ...
197    2320
198    1515
199    2230
200
201    1940
Name: TIME_OCCURRED, Length: 202, dtype: object
```

PATROL DIVISION: These numbers belong to the respective precincts, they are float values, would be converted to integers Same is the case for **CRIME_CODE**

In [55]:
```
data['PATROL_DIVISION'] = data['PATROL_DIVISION'].astype('Int64')
data['CRIME_CODE'] = data['CRIME_CODE'].astype('Int64')
```

In [56]:
```
data.head()
```

Out[56]:

	DATE_REPORTED	DATE_OCCURRED	TIME_OCCURRED	PATROL_DIVISION	AREA_NAME	CRI
0	2008-04-10	2008-04-10	1910	73	BROOKLYN	
1	2007-06-03	2007-06-03	1523	28	MANHATTAN	
2	2010-02-16	2010-02-16	2050	102	QUEENS	
3	2009-11-10	2009-11-10	1635	79	BROOKLYN	
4	2006-04-25	2006-04-11	0930	123	STATEN ISLAND	

VICTIM_AGE was converted to range values, as NY database consisted of range and LA database consisted to absolute values

In [57]:

```
def range_from_age(age):
    if age==0:
        return "UNKNOWN"
    if age<18:
        return "<18"
    if age > 18 and age <=24:
        return "18-24"
    if age >24 and age <=44:
        return "25-44"
    if age >44 and age <=65:
        return "45-65"
    else:
        return "65+"

for index,val in enumerate(data["VICTIM_AGE"]):
    if type(val)==str:
        if str.isdigit(val):
            data["VICTIM_AGE"][index]=range_from_age(int(val))
    else:
        data["VICTIM_AGE"][index]="UNKNOWN"




data["VICTIM_AGE"]
```

Out[57]:

```
0        18-24
1      UNKNOWN
2      UNKNOWN
3      UNKNOWN
4        25-44
        ...
197    UNKNOWN
198      45-65
199      18-24
200      25-44
201      18-24
Name: VICTIM_AGE, Length: 202, dtype: object
```

VICTIM_SEX has a defaut value for unknown which is "D" , the current default value is "E" for unknown

In [58]:

```
for index,val in enumerate(data["VICTIM_SEX"]):
    if val =="D":
        data["VICTIM_SEX"][index]="E"
```

In [59]:
```
data["VICTIM_SEX"]
```

Out[59]:
```
0      M
1      E
2      E
3      E
4      M
      ..
197    M
198    M
199    M
200    F
201    M
Name: VICTIM_SEX, Length: 202, dtype: object
```

VICTIM_RACE is menioned in words for NY dataset whereas LA dataset provides a character to word mapping, hence exapnding the map to get appropriate race

In [60]:
```
char_to_descent_map={"A":"Other Asian", "B": "Black", "C" : "Chinese",
"D" : "Cambodian", "F": "Filipino","G" :"Guamanian" ,"H":"Hispanic/Lati
n/Mexican" ,"I": "American Indian/Alaskan Native" ,"J": "Japanese", "K":
"Korean", "L" :"Laotian" ,"O" : "Other", "P": "Pacific Islander", "S":
"Samoan", "U": "Hawaiian","V": "Vietnamese", "W" : "White" ,"X": "Unknow
n" ,"Z" :"Asian Indian"}

for index,string in enumerate(data["VICTIM_RACE"]):
    if len(string.lstrip().rstrip())==1:
        data["VICTIM_RACE"][index]=char_to_descent_map[string]
```

In [61]:
```
data["VICTIM_RACE"]
```

Out[61]:
```
0                        BLACK
1                      UNKNOWN
2                      UNKNOWN
3                      UNKNOWN
4                        WHITE
                 ...
197                      Other
198    Hispanic/Latin/Mexican
199    Hispanic/Latin/Mexican
200                      Other
201    Hispanic/Latin/Mexican
Name: VICTIM_RACE, Length: 202, dtype: object
```

Drop missing values

In [62]:
```
data = data.dropna(how='any',axis=0)
```

In [63]:
```
data.head()
```

Out[63]:

	DATE_REPORTED	DATE_OCCURRED	TIME_OCCURRED	PATROL_DIVISION	AREA_NAME	CRI
0	2008-04-10	2008-04-10	1910	73	BROOKLYN	
1	2007-06-03	2007-06-03	1523	28	MANHATTAN	
2	2010-02-16	2010-02-16	2050	102	QUEENS	
3	2009-11-10	2009-11-10	1635	79	BROOKLYN	
5	2011-06-24	2011-06-23	2030	81	BROOKLYN	

Saving the data to a file

In [64]:
```
data.to_csv("cleaned_data.csv")
```

In [48]:
```

```

In [65]:
```
data
```

Out[65]:

	DATE_REPORTED	DATE_OCCURRED	TIME_OCCURRED	PATROL_DIVISION	AREA_NAME	C
0	2008-04-10	2008-04-10	1910	73	BROOKLYN	
1	2007-06-03	2007-06-03	1523	28	MANHATTAN	
2	2010-02-16	2010-02-16	2050	102	QUEENS	
3	2009-11-10	2009-11-10	1635	79	BROOKLYN	
5	2011-06-24	2011-06-23	2030	81	BROOKLYN	
...	...	...	...	...	...	
197	2010-04-01	2010-03-29	2320	1	Central	
198	2010-03-31	2010-03-31	1515	1	Central	
199	2010-04-03	2010-04-02	2230	1	Central	
200	2010-04-03	2010-04-03		1	Central	
201	2010-04-06	2010-04-05	1940	1	Central	

164 rows × 13 columns

In [ ]:
```
```