HM1_py

February 3, 2025

1 Homework N1

1.1 Part 1: Data Cleaning and Exploration

1.1.1 1) Load the dataset. Check the first 5 rows

[114]:
```
import pandas as pd
import numpy as np
df = pd.read_csv("crime_data.csv")
df.head(5)
```

[114]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	\
0	241711715	08/01/2024 12:00:00 AM	08/01/2024 12:00:00 AM	1319	17	
1	231014031	09/21/2023 12:00:00 AM	09/15/2023 12:00:00 AM	1930	10	
2	231010808	06/27/2023 12:00:00 AM	06/26/2023 12:00:00 AM	1230	10	
3	211410441	04/25/2021 12:00:00 AM	04/25/2021 12:00:00 AM	2330	14	
4	211114569	10/25/2021 12:00:00 AM	10/25/2021 12:00:00 AM	1455	11	

	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	\
0	Devonshire	1791	1	440	
1	West Valley	1011	2	354	
2	West Valley	1015	2	354	
3	Pacific	1488	2	626	
4	Northeast	1123	1	210	

	Crm Cd Desc	…	Status	Status Desc	Crm Cd 1	\
0	THEFT PLAIN - PETTY ($950 & UNDER)	…	IC	Invest Cont	440.0	
1	THEFT OF IDENTITY	…	IC	Invest Cont	354.0	
2	THEFT OF IDENTITY	…	IC	Invest Cont	354.0	
3	INTIMATE PARTNER - SIMPLE ASSAULT	…	IC	Invest Cont	626.0	
4	ROBBERY	…	IC	Invest Cont	210.0	

	Crm Cd 2	Crm Cd 3	Crm Cd 4			LOCATION	\
0	NaN	NaN	NaN	8300	KELVIN	AV	
1	NaN	NaN	NaN	18900	CANTLAY	ST	
2	NaN	NaN	NaN	7300	ENFIELD	AV	
3	NaN	NaN	NaN	5800 W	CENTURY	BL	
4	NaN	NaN	NaN	2900	LOS FELIZ	BL	

```
  Cross Street      LAT       LON
0          NaN  34.2200 -118.5863
1          NaN  34.2023 -118.5458
2          NaN  34.2033 -118.5241
3          NaN  33.9456 -118.3835
4          NaN   0.0000    0.0000

[5 rows x 28 columns]
```

1.1.2 2) Identify columns with missing values and their respective counts. Drop columns where more than 50% of the data is missing (store this version as a new dataset).

[115]:
```
missing_values_cols = df.columns[df.isnull().sum() != 0]
counts = {}
for col in list(missing_values_cols):
    counts[col] = df[col].isna().sum()
display(counts)

new_df = df.dropna(thresh=len(df.index) // 2, axis=1)
new_df.head()
```

```
{'Mocodes': 7498,
 'Vict Sex': 7163,
 'Vict Descent': 7165,
 'Premis Desc': 29,
 'Weapon Used Cd': 33654,
 'Weapon Desc': 33654,
 'Crm Cd 1': 2,
 'Crm Cd 2': 46448,
 'Crm Cd 3': 49885,
 'Crm Cd 4': 49995,
 'Cross Street': 42258}
```

[115]:
```
       DR_NO               Date Rptd                 DATE OCC  TIME OCC  AREA  \
0  241711715  08/01/2024 12:00:00 AM  08/01/2024 12:00:00 AM      1319    17
1  231014031  09/21/2023 12:00:00 AM  09/15/2023 12:00:00 AM      1930    10
2  231010808  06/27/2023 12:00:00 AM  06/26/2023 12:00:00 AM      1230    10
3  211410441  04/25/2021 12:00:00 AM  04/25/2021 12:00:00 AM      2330    14
4  211114569  10/25/2021 12:00:00 AM  10/25/2021 12:00:00 AM      1455    11

     AREA NAME  Rpt Dist No  Part 1-2  Crm Cd  \
0   Devonshire         1791         1     440
1  West Valley         1011         2     354
2  West Valley         1015         2     354
3      Pacific         1488         2     626
```

```
4    Northeast         1123         1     210

                          Crm Cd Desc  … Vict Sex  Vict Descent Premis Cd  \
0  THEFT PLAIN - PETTY ($950 & UNDER)  …        M             O     501.0
1                   THEFT OF IDENTITY  …        F             W     501.0
2                   THEFT OF IDENTITY  …        F             O     501.0
3   INTIMATE PARTNER - SIMPLE ASSAULT  …        F             B     503.0
4                             ROBBERY  …        X             X     412.0

                                Premis Desc  Status  Status Desc Crm Cd 1  \
0                    SINGLE FAMILY DWELLING      IC  Invest Cont    440.0
1                    SINGLE FAMILY DWELLING      IC  Invest Cont    354.0
2                    SINGLE FAMILY DWELLING      IC  Invest Cont    354.0
3                                     HOTEL      IC  Invest Cont    626.0
4  ELECTRONICS STORE (IE:RADIO SHACK, ETC.)      IC  Invest Cont    210.0

                                  LOCATION      LAT       LON
0   8300    KELVIN                      AV  34.2200 -118.5863
1  18900    CANTLAY                     ST  34.2023 -118.5458
2   7300    ENFIELD                     AV  34.2033 -118.5241
3   5800 W  CENTURY                     BL  33.9456 -118.3835
4   2900    LOS FELIZ                   BL   0.0000    0.0000

[5 rows x 22 columns]
```

1.1.3 3) Convert the DATE OCC column to a datetime format. Extract the year, month, and day into separate columns. Create a new column for the hour using the TIME OCC column.

[116]:
```
new_df["DATE OCC"] = pd.to_datetime(new_df["DATE OCC"])
new_df["Month"] = new_df["DATE OCC"].dt.month
new_df["Year"] = new_df["DATE OCC"].dt.year
new_df["Day"] = new_df["DATE OCC"].dt.day
new_df["Hour"] = new_df["TIME OCC"].apply(lambda x: str(x)[:-2]) #Taking the␣
 ↪num before the last two
new_df["Hour"] = new_df["Hour"].apply(lambda x: int(x) if x else 0) #If the␣
 ↪string is empty than it's 12pm
new_df.head()
```

```
C:\Users\Hovgr\AppData\Local\Temp\ipykernel_19868\4131287626.py:1: UserWarning:
Could not infer format, so each element will be parsed individually, falling
back to `dateutil`. To ensure parsing is consistent and as-expected, please
specify a format.
  new_df["DATE OCC"] = pd.to_datetime(new_df["DATE OCC"])
C:\Users\Hovgr\AppData\Local\Temp\ipykernel_19868\4131287626.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df["DATE OCC"] = pd.to_datetime(new_df["DATE OCC"])
C:\Users\Hovgr\AppData\Local\Temp\ipykernel_19868\4131287626.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df["Month"] = new_df["DATE OCC"].dt.month
C:\Users\Hovgr\AppData\Local\Temp\ipykernel_19868\4131287626.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df["Year"] = new_df["DATE OCC"].dt.year
C:\Users\Hovgr\AppData\Local\Temp\ipykernel_19868\4131287626.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df["Day"] = new_df["DATE OCC"].dt.day
C:\Users\Hovgr\AppData\Local\Temp\ipykernel_19868\4131287626.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df["Hour"] = new_df["TIME OCC"].apply(lambda x: str(x)[:-2]) #Taking the
num before the last two
C:\Users\Hovgr\AppData\Local\Temp\ipykernel_19868\4131287626.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df["Hour"] = new_df["Hour"].apply(lambda x: int(x) if x else 0) #If the
string is empty than it's 12pm
```

[116]:
```
       DR_NO               Date Rptd   DATE OCC  TIME OCC  AREA    AREA NAME  \
0  241711715  08/01/2024 12:00:00 AM 2024-08-01      1319    17   Devonshire
1  231014031  09/21/2023 12:00:00 AM 2023-09-15      1930    10  West Valley
2  231010808  06/27/2023 12:00:00 AM 2023-06-26      1230    10  West Valley
3  211410441  04/25/2021 12:00:00 AM 2021-04-25      2330    14      Pacific
4  211114569  10/25/2021 12:00:00 AM 2021-10-25      1455    11    Northeast

   Rpt Dist No  Part 1-2  Crm Cd                          Crm Cd Desc  …  \
0         1791         1     440  THEFT PLAIN - PETTY ($950 & UNDER)  …
1         1011         2     354                   THEFT OF IDENTITY  …
2         1015         2     354                   THEFT OF IDENTITY  …
3         1488         2     626   INTIMATE PARTNER - SIMPLE ASSAULT  …
4         1123         1     210                             ROBBERY  …

  Status  Status Desc Crm Cd 1                                  LOCATION  \
0     IC  Invest Cont    440.0   8300    KELVIN                      AV
1     IC  Invest Cont    354.0  18900    CANTLAY                     ST
2     IC  Invest Cont    354.0   7300    ENFIELD                     AV
3     IC  Invest Cont    626.0   5800 W  CENTURY                     BL
4     IC  Invest Cont    210.0   2900    LOS FELIZ                   BL

       LAT       LON Month  Year  Day Hour
0  34.2200 -118.5863     8  2024    1   13
1  34.2023 -118.5458     9  2023   15   19
2  34.2033 -118.5241     6  2023   26   12
3  33.9456 -118.3835     4  2021   25   23
4   0.0000    0.0000    10  2021   25   14

[5 rows x 26 columns]
```

1.1.4 4) Filter the dataset for crimes that occurred in 2023. Further filter crimes with the description BURGLARY in the Crm Cd Desc column.

[117]:
```
df_2023 = new_df[new_df["Year"] == 2023]
df_bulg_2023 = df_2023[df_2023["Crm Cd Desc"] == "BURGLARY"]
df_bulg_2023.head()
```

[117]:
```
         DR_NO               Date Rptd   DATE OCC  TIME OCC  AREA  \
47   231107877  04/15/2023 12:00:00 AM 2023-01-15       500    11
147  231912840  08/15/2023 12:00:00 AM 2023-08-14      2200    19
292  230813484  08/19/2023 12:00:00 AM 2023-08-19       510     8
317  230126836  12/20/2023 12:00:00 AM 2023-12-10      1200     1
326  231506351  02/26/2023 12:00:00 AM 2023-02-22      1230    15

       AREA NAME  Rpt Dist No  Part 1-2  Crm Cd Crm Cd Desc  … Status  \
47     Northeast         1151         1     310    BURGLARY  …     IC
147      Mission         1962         1     310    BURGLARY  …     IC
```

```
292      West LA          857         1     310    BURGLARY  …      IC
317      Central          154         1     310    BURGLARY  …      IC
326  N Hollywood         1562         1     310    BURGLARY  …      IC

     Status Desc Crm Cd 1                                LOCATION      LAT  \
47    Invest Cont    310.0   5000 W  SUNSET                    BL  34.0981
147   Invest Cont    310.0  15000    CORE                      LN  34.2424
292   Invest Cont    310.0   9400 W  PICO                      BL  34.0553
317   Invest Cont    310.0    100 E  6TH                       ST  34.0460
326   Invest Cont    310.0   4400    BABCOCK                   AV  34.1504

           LON Month  Year  Day Hour
47  -118.2983     1  2023   15    5
147 -118.4596     8  2023   14   22
292 -118.3943     8  2023   19    5
317 -118.2493    12  2023   10   12
326 -118.4063     2  2023   22   12

[5 rows x 26 columns]
```

1.1.5 5) Group the data by AREA NAME and calculate the total number of crimes and the average victim age. Sort the results by total crimes in descending order.

[118]:
```
df_bulg_2023.groupby("AREA NAME").agg({"Vict Age": ['count', 'mean']}).
↪sort_values(by=("Vict Age", "count"), ascending=False)
```

[118]:

	Vict Age	
	count	mean
AREA NAME		
Devonshire	58	43.896552
West LA	58	40.879310
Olympic	53	30.283019
West Valley	52	35.173077
N Hollywood	50	29.660000
Wilshire	48	39.041667
Pacific	48	24.520833
Van Nuys	44	41.704545
Southwest	32	35.656250
77th Street	31	30.129032
Topanga	30	42.300000
Central	29	22.241379
Rampart	26	24.653846
Hollywood	25	28.240000
Northeast	25	32.120000
Newton	24	20.333333
Hollenbeck	22	19.318182
Harbor	22	27.818182

```
Foothill          18  35.888889
Southeast         16  40.750000
Mission           14  42.571429
```

1.2 Part 2: Further Exploration

1.2.1 1) Find the top 3 most frequent Crm Cd Desc values.

[119]:
```
import pandas as pd
import numpy as np
df = pd.read_csv("crime_data.csv")
df.head(5)
```

[119]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA
0	241711715	08/01/2024 12:00:00 AM	08/01/2024 12:00:00 AM	1319	17
1	231014031	09/21/2023 12:00:00 AM	09/15/2023 12:00:00 AM	1930	10
2	231010808	06/27/2023 12:00:00 AM	06/26/2023 12:00:00 AM	1230	10
3	211410441	04/25/2021 12:00:00 AM	04/25/2021 12:00:00 AM	2330	14
4	211114569	10/25/2021 12:00:00 AM	10/25/2021 12:00:00 AM	1455	11

	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd
0	Devonshire	1791	1	440
1	West Valley	1011	2	354
2	West Valley	1015	2	354
3	Pacific	1488	2	626
4	Northeast	1123	1	210

	Crm Cd Desc	…	Status	Status Desc	Crm Cd 1
0	THEFT PLAIN - PETTY ($950 & UNDER)	…	IC	Invest Cont	440.0
1	THEFT OF IDENTITY	…	IC	Invest Cont	354.0
2	THEFT OF IDENTITY	…	IC	Invest Cont	354.0
3	INTIMATE PARTNER - SIMPLE ASSAULT	…	IC	Invest Cont	626.0
4	ROBBERY	…	IC	Invest Cont	210.0

	Crm Cd 2	Crm Cd 3	Crm Cd 4		LOCATION
0	NaN	NaN	NaN	8300 KELVIN	AV
1	NaN	NaN	NaN	18900 CANTLAY	ST
2	NaN	NaN	NaN	7300 ENFIELD	AV
3	NaN	NaN	NaN	5800 W CENTURY	BL
4	NaN	NaN	NaN	2900 LOS FELIZ	BL

	Cross Street	LAT	LON
0	NaN	34.2200	-118.5863
1	NaN	34.2023	-118.5458
2	NaN	34.2033	-118.5241
3	NaN	33.9456	-118.3835
4	NaN	0.0000	0.0000

```
[5 rows x 28 columns]
```

[120]:
```
df.groupby("Crm Cd Desc")["Crm Cd Desc"].count().sort_values(ascending=False)[:3]
```

[120]:
```
Crm Cd Desc
VEHICLE - STOLEN            5733
BATTERY - SIMPLE ASSAULT    3715
THEFT OF IDENTITY           3169
Name: Crm Cd Desc, dtype: int64
```

1.2.2 2) Group the data by Hour and count the number of crimes.

[121]:
```
df["Hour"] = df["TIME OCC"].apply(lambda x: str(x)[:-2]) #Taking the num before the last two
df["Hour"] = df["Hour"].apply(lambda x: int(x) if x else 0) #If the string is empty than it's 12pm
df.groupby("Hour")["Hour"].count()
```

[121]:
```
Hour
0     2025
1     1490
2     1241
3     1063
4      975
5      848
6     1106
7     1302
8     1776
9     1841
10    2110
11    2114
12    3446
13    2246
14    2504
15    2667
16    2597
17    2914
18    3015
19    2762
20    2830
21    2553
22    2469
23    2106
Name: Hour, dtype: int64
```

1.2.3 3) Group the data by Vict Sex and calculate: Total crimes, Average victim age.

[122]:
```
df.groupby("Vict Sex").agg({"Vict Age": ['count', 'mean']})
```

[122]:

	Vict Age	
	count	mean
Vict Sex		
F	17922	38.164156
H	3	36.333333
M	20076	37.165621
X	4836	2.672043

1.3 Part 4: Advanced Analysis

1.3.1 Create a new column, Severity Score, based on the following rules:

- Assign 5 points if a weapon was used (Weapon Used Cd is not null).
- Assign 3 points for crimes under BURGLARY.
- Assign 1 point for all other crimes.
- Group by AREA NAME and find the total severity score for each area.

[123]:
```
import pandas as pd
import numpy as np
df = pd.read_csv("crime_data.csv")
```

[162]:
```
df["Severity Score"] = df["Weapon Used Cd"].apply(lambda x: 5 if not pd.isna(x)␣
 ↪else 0)
df["Severity Score"] += df["Crm Cd Desc"].apply(lambda x: 3 if x == "BURGLARY"␣
 ↪else 0)
df["Severity Score"] = df["Severity Score"].apply(lambda x: 1 if x == 0 else x)

df.groupby("AREA NAME")["Severity Score"].sum()
```

[162]:
```
AREA NAME
77th Street    9298
Central        8625
Devonshire     4510
Foothill       3856
Harbor         4949
Hollenbeck     4454
Hollywood      6707
Mission        4499
N Hollywood    5476
Newton         6815
Northeast      4476
Olympic        6275
Pacific        6652
Rampart        6267
```

```
Southeast      7162
Southwest      7029
Topanga        4541
Van Nuys       4751
West LA        4763
West Valley    4994
Wilshire       5212
Name: Severity Score, dtype: int64
```

1.4 Bonus Part:

[172]:
```
def distance(lat1, lon1, lat2, lon2): #Get distance between two points
    r = 6371 # km
    p = np.pi / 180

    a = 0.5 - np.cos((lat2-lat1)*p)/2 + np.cos(lat1*p) * np.cos(lat2*p) * (1-np.
 ↪cos((lon2-lon1)*p))/2
    return 2 * r * np.arcsin(np.sqrt(a))

max_dist = 5 #maximum distance in which two points are considered in the same␣
 ↪area

ref_lat = df.iloc[0].LAT
ref_lon = df.iloc[0].LON

df["distance_from_ref"] = distance(ref_lat, ref_lon, df["LAT"], df["LON"])

df[df["distance_from_ref"] < max_dist]
```

[172]:

	DR_NO	Date Rptd	DATE OCC	TIME OCC	\
0	241711715	08/01/2024 12:00:00 AM	08/01/2024 12:00:00 AM	1319	
1	231014031	09/21/2023 12:00:00 AM	09/15/2023 12:00:00 AM	1930	
17	201710780	07/05/2020 12:00:00 AM	07/05/2020 12:00:00 AM	1120	
41	241708301	04/13/2024 12:00:00 AM	10/30/2023 12:00:00 AM	2130	
67	212107949	04/20/2021 12:00:00 AM	04/20/2021 12:00:00 AM	800	
…	…	…	…	…	
49877	221017153	12/01/2022 12:00:00 AM	11/17/2022 12:00:00 AM	740	
49919	211004241	01/09/2021 12:00:00 AM	01/06/2021 12:00:00 AM	1600	
49961	201709510	05/28/2020 12:00:00 AM	05/28/2020 12:00:00 AM	600	
49977	202115969	11/14/2020 12:00:00 AM	11/14/2020 12:00:00 AM	1345	
49992	202110106	06/02/2020 12:00:00 AM	05/29/2020 12:00:00 AM	1600	

	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	\
0	17	Devonshire	1791	1	440	
1	10	West Valley	1011	2	354	
17	17	Devonshire	1794	2	625	
41	17	Devonshire	1782	1	445	

```
67       21      Topanga         2157         1     761
…     …          …           …       …     …
49877    10  West Valley         1003         2     624
49919    10  West Valley         1001         1     510
49961    17   Devonshire         1783         1     510
49977    21      Topanga         2146         2     930
49992    21      Topanga         2136         1     330

                                  Crm Cd Desc  … Crm Cd 1  Crm Cd 2  \
0           THEFT PLAIN - PETTY ($950 & UNDER)  …    440.0       NaN
1                             THEFT OF IDENTITY  …    354.0       NaN
17                               OTHER ASSAULT  …    625.0       NaN
41          DISHONEST EMPLOYEE ATTEMPTED THEFT  …    445.0       NaN
67                             BRANDISH WEAPON  …    761.0     998.0
…                                         …  …      …       …
49877                 BATTERY - SIMPLE ASSAULT  …    624.0       NaN
49919                         VEHICLE - STOLEN  …    510.0       NaN
49961                         VEHICLE - STOLEN  …    510.0       NaN
49977  CRIMINAL THREATS - NO WEAPON DISPLAYED  …    930.0       NaN
49992                    BURGLARY FROM VEHICLE  …    330.0       NaN

      Crm Cd 3 Crm Cd 4                                  LOCATION  \
0          NaN      NaN   8300    KELVIN                     AV
1          NaN      NaN  18900    CANTLAY                    ST
17         NaN      NaN   8300    RESEDA                     BL
41         NaN      NaN   8800    CORBIN                     AV
67         NaN      NaN                                DE SOTO
…        …      …                                     …
49877      NaN      NaN           ROSCOE                     BL
49919      NaN      NaN   7700    TAMPA                      AV
49961      NaN      NaN   9000    VANALDEN                   AV
49977      NaN      NaN  21500    BASSETT                    ST
49992      NaN      NaN   7200    OWENSMOUTH                 AV

                         Cross Street      LAT       LON Severity Score  \
0                                 NaN  34.2200 -118.5863              1
1                                 NaN  34.2023 -118.5458              1
17                                NaN  34.2208 -118.5361              5
41                                NaN  34.2302 -118.5623              1
67                            VANOWEN  34.1938 -118.5884              5
…                                   …       …       …             …
49877  RESEDA                      BL  34.2207 -118.5361              5
49919                             NaN  34.2102 -118.5535              1
49961                             NaN  34.2341 -118.5493              1
49977                             NaN  34.1956 -118.6003              5
49992                             NaN  34.2010 -118.6016              1
```

```
      distance_from_ref
0              0.000000
1              4.212257
17             4.616491
41             2.480981
67             2.919701
…                 …
49877          4.616293
49919          3.206808
49961          3.745619
49977          3.003110
49992          2.538296

[2623 rows x 30 columns]
```