

NE_inmates_offenses_1

March 8, 2021

0.1 Dataset

The data being used was from inmate databases acquired from the Nebraska Department of Corrections Public Records. https://dcs-inmatesearch.ne.gov/Corrections/COR_download.htm

The objective is to combine the active and complete databases from the Nebraska Department of Corrections, delete unnecessary rows, and prepare the new dataset for analysis

0.2 1. Load the Data

Load the data using Pandas.

Pandas 'ExcelFile' will load the data as a Pandas DataFrame object. The Dataset has two pages, and they are assigned their own DataFrame.

```
[1]: import numpy as np
import pandas as pd
```

```
[2]: xls = pd.ExcelFile('inmateDB_updated.xlsx')
df1 = pd.read_excel(xls, 'Record Type 1')
df2 = pd.read_excel(xls, 'Record Type 2')
```

0.3 2. Check the First Few Rows of Data

The DataFrame's first five rows can be viewed using the .head() method

```
[3]: df1.head()
```

```
[3]:  ID NUMBER COMMITTED LAST NAME FIRST NAME MIDDLE NAME NAME EXTENSION \
0      1702          CLIFFORD   BRADLEY      NaN
1      6145           KANE     THOMAS      NaN
2      6452          ATKINS    LARRY      NaN
3     12444     SHANEYFELT   CHARLEY      NaN
4     15379          BEADES      JOE      NaN

    LEGAL LAST NAME FIRST NAME2 MIDDLE NAME3 NAME EXTENSION4 DATE OF BIRTH ... \
0      NaN      NaN      NaN      NaN      NaN      NaT ...
1      NaN      NaN      NaN      NaN      NaN  1928-12-21 ...
2      NaN      NaN      NaN      NaN      NaN  1929-07-26 ...
3      NaN      NaN      NaN      NaN      NaN  1905-04-10 ...
```

| | | | | | | |
|---|-----|-----|-----|-----|------------|-----|
| 4 | NaN | NaN | NaN | NaN | 1924-10-12 | ... |
|---|-----|-----|-----|-----|------------|-----|

| | PAROLE ELIGIBILITY DATE | EARLIEST POSSIBLE RELEASE DATE | \ |
|---|-------------------------|--------------------------------|---|
| 0 | NaN | NaN | |
| 1 | 1952-06-20 00:00:00 | NaN | |
| 2 | NaN | NaN | |
| 3 | NaN | NaN | |
| 4 | 1955-05-02 00:00:00 | LFE | |

| | GOOD TIME LAW INST | RELEASE DATE | \ |
|---|--------------------|--------------|---|
| 0 | | 1986-01-06 | |
| 1 | 2926 | 1952-08-31 | |
| 2 | | 1955-07-20 | |
| 3 | | 1987-12-24 | |
| 4 | 2926 | 1989-07-19 | |

| | INST RELEASE TYPE | PAROLE BOARD | NEXT REVIEW DATE(MONTH&YEAR) | \ |
|---|----------------------|--------------|------------------------------|---|
| 0 | MANDATORY DISCHARGE | | NaN | |
| 1 | ESCAPE | | NaN | |
| 2 | DISCRETIONARY PAROLE | | NaN | |
| 3 | MANDATORY DISCHARGE | | NaN | |
| 4 | DISCRETIONARY PAROLE | | NaN | |

| | PAROLE BOARD FINAL HEARING DATE(MONTH&YEAR) | PAROLE BOARD STATUS | \ |
|---|---|---------------------|---|
| 0 | NaT | NaN | |
| 1 | NaT | NaN | |
| 2 | NaT | PAROLED | |
| 3 | NaT | NaN | |
| 4 | NaT | PAROLED | |

| | PAROLE DATE | PAROLE DISCHARGE | DESC |
|---|-------------|-------------------------------|------|
| 0 | NaT | | NaN |
| 1 | NaT | | |
| 2 | 1980-12-09 | EARLY DISCHARGE BY PAROLE BRD | |
| 3 | NaT | | |
| 4 | 1993-01-17 | EARLY DISCHARGE BY PAROLE BRD | |

[5 rows x 32 columns]

```
[4]: df2.head()
```

| | ID | ID NUMBER | OFFENSE | MINIMUM YEAR OR TERM | MINIMUM MONTH | MINIMUM DAY | \ |
|---|----|-----------|---------|----------------------|---------------|-------------|---|
| 0 | 1 | 6145 | | 1 | 0.0 | 0.0 | |
| 1 | 2 | 6452 | | 2 | 0.0 | 0.0 | |
| 2 | 3 | 12444 | | 1 | 0.0 | 0.0 | |
| 3 | 4 | 15379 | | 10 | 0.0 | 0.0 | |
| 4 | 5 | 15410 | | 2 | 0.0 | 0.0 | |

| | OFFENSE | MAXIMUM YEAR OR TERM | MAXIMUM MONTH | MAXIMUM DAY | \ |
|---|---------|----------------------|---------------|-------------|---|
| 0 | | 3 | 0.0 | 0.0 | |
| 1 | | 10 | 0.0 | 0.0 | |
| 2 | | 9 | 0.0 | 0.0 | |
| 3 | | LFE | NaN | NaN | |
| 4 | | 3 | 0.0 | 0.0 | |

| | OFFENSE | ARREST DESC | FELONY | MSDMNR CODE | OFFENSE TYPE CODE | \ |
|---|--------------------|-------------|--------|-------------|-------------------|---|
| 0 | CATTLE STEALING | | FELONY | | * | |
| 1 | FORGERY 2ND DEGREE | | NaN | | * | |
| 2 | BURGLARY | | FELONY | | * | |
| 3 | MURDER 2ND DEGREE | | NaN | | * | |
| 4 | FORGERY 1ST DEGREE | | FELONY | | * | |

| | OFFENSE | ATTEMPT DESC | HABITUAL | CRIMINAL | OFFENSE RUN CODE | COUNTY COMMITTED | \ |
|---|---------|--------------|----------|----------|------------------|------------------|---|
| 0 | | NaN | | NaN | CC | MADISON | |
| 1 | | NaN | | NaN | CC | SCOTTS BLUFF | |
| 2 | | NaN | | NaN | CC | HAMILTON | |
| 3 | | NaN | | NaN | CC | DOUGLAS | |
| 4 | | NaN | | NaN | CC | SCOTTS BLUFF | |

| | Offense | Arrest CD | Offense | Arrest |
|---|---------|-----------|--------------------|----------|
| 0 | | D21 | | THEFT |
| 1 | | E02 | FORGERY 2ND DEGREE | |
| 2 | | D11 | | BURGLARY |
| 3 | | B02 | MURDER 2ND DEGREE | |
| 4 | | E01 | FORGERY 1ST DEGREE | |

0.4 3. Description of Data

The DataFrame `info()` method is used to see helpful descriptions of the data, such as the column name and number of rows. The ‘Non-Null Count’ is the number of rows that have a value for that particular column. The ‘Dtype’ is the data type found within each column. An `int64` is an integer, an object type is usually written text, and `datetime64` is a date time value.

```
[5]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72954 entries, 0 to 72953
Data columns (total 32 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                            72954 non-null  int64
1   COMMITTED LAST NAME                  72953 non-null  object
2   FIRST NAME                           72953 non-null  object
3   MIDDLE NAME                          54905 non-null  object
4   NAME EXTENSION                       72954 non-null  object
```

| | | | |
|----|---|----------------|----------------|
| 5 | LEGAL LAST NAME | 1045 non-null | object |
| 6 | FIRST NAME2 | 1045 non-null | object |
| 7 | MIDDLE NAME3 | 1045 non-null | object |
| 8 | NAME EXTENSION4 | 1045 non-null | object |
| 9 | DATE OF BIRTH | 72940 non-null | datetime64[ns] |
| 10 | RACE DESC | 72954 non-null | object |
| 11 | GENDER | 72954 non-null | object |
| 12 | FACILITY | 14080 non-null | object |
| 13 | CURRENT SENTENCE PARDONED OR COMMUTED DATE | 72954 non-null | object |
| 14 | GUN CLAUSE | 256 non-null | object |
| 15 | SENTENCE BEGIN DATE | 71475 non-null | datetime64[ns] |
| 16 | MIN TERM/YEAR | 72954 non-null | object |
| 17 | MIN MONTH | 72588 non-null | float64 |
| 18 | MIN DAY | 72588 non-null | float64 |
| 19 | MAX TERM/YEAR | 72954 non-null | object |
| 20 | MAX MONTH | 72383 non-null | float64 |
| 21 | MAX DAY | 72383 non-null | float64 |
| 22 | PAROLE ELIGIBILITY DATE | 66996 non-null | object |
| 23 | EARLIEST POSSIBLE RELEASE DATE | 72306 non-null | object |
| 24 | GOOD TIME LAW | 72954 non-null | object |
| 25 | INST RELEASE DATE | 67994 non-null | datetime64[ns] |
| 26 | INST RELEASE TYPE | 67994 non-null | object |
| 27 | PAROLE BOARD NEXT REVIEW DATE(MONTH&YEAR) | 41546 non-null | object |
| 28 | PAROLE BOARD FINAL HEARING DATE(MONTH&YEAR) | 1973 non-null | datetime64[ns] |
| 29 | PAROLE BOARD STATUS | 69868 non-null | object |
| 30 | PAROLE DATE | 25380 non-null | datetime64[ns] |
| 31 | PAROLE DISCHARGE DESC | 34862 non-null | object |

dtypes: datetime64[ns](5), float64(4), int64(1), object(22)

memory usage: 17.8+ MB

```
[6]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129897 entries, 0 to 129896
Data columns (total 17 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID                                    129897 non-null  int64
1   ID NUMBER                            129897 non-null  int64
2   OFFENSE MINIMUM YEAR OR TERM         129897 non-null  object
3   MINIMUM MONTH                        129418 non-null  float64
4   MINIMUM DAY                          129418 non-null  float64
5   OFFENSE MAXIMUM YEAR OR TERM         129897 non-null  object
6   MAXIMUM MONTH                        129181 non-null  float64
7   MAXIMUM DAY                          129181 non-null  float64
8   OFFENSE ARREST DESC                  129897 non-null  object
9   FELONY MSDMNR CODE                   115048 non-null  object
10  OFFENSE TYPE CODE                    129897 non-null  object
```

```

11 OFFENSE ATTEMPT DESC          16430 non-null  object
12 HABITUAL CRIMINAL            767 non-null   object
13 OFFENSE RUN CODE             105476 non-null object
14 COUNTY COMMITTED             129869 non-null object
15 Offense Arrest CD            129826 non-null object
16 Offense Arrest                129826 non-null object
dtypes: float64(4), int64(2), object(11)
memory usage: 16.8+ MB

```

0.5 4. Creating a New Active Prisoner Distinction Column

A new column will be created that will reflect if the inmates are actively incarcerated or are no longer in prison. This will be created by assigning a new column that includes inmates in both the full and active inmate databases

0.5.1 4.1 Examining the Active Prisoner Database

The Active prisoner database will be loaded and examined in the same manner as with the full inmate database

```
[7]: xls = pd.ExcelFile('inmateDownloadActive.xlsx')
df_Active = pd.read_excel(xls, 'Record Type 1')
```

Some of the Active prisoner database columns are slightly different than the full database, but it contains the same basic information as the full database.

```
[8]: df_Active.head()
```

```
[8]:
```

| | ID NUMBER | COMMITTED | LAST NAME | FIRST NAME | MIDDLE NAME | NAME EXTENSION | \ |
|---|-----------|-----------|-----------|------------|-------------|----------------|---|
| 0 | 6145 | | KANE | THOMAS | | NaN | |
| 1 | 20841 | | ARNOLD | WILLIAM | | L | |
| 2 | 25324 | | WALKER | RICHARD | | T | |
| 3 | 25565 | | ALVAREZ | THOMAS | | A | |
| 4 | 26103 | | ADAMS | BRIAN | | J | |

| | LEGAL LAST NAME | FIRST NAME.1 | MIDDLE NAME.1 | NAME EXTENSION.1 | DATE OF BIRTH | \ |
|---|-----------------|--------------|---------------|------------------|---------------|---|
| 0 | NaN | NaN | NaN | NaN | 1928-12-21 | |
| 1 | NaN | NaN | NaN | NaN | 1942-08-28 | |
| 2 | NaN | NaN | NaN | NaN | 1946-12-24 | |
| 3 | NaN | NaN | NaN | NaN | 1947-10-24 | |
| 4 | NaN | NaN | NaN | NaN | 1949-04-20 | |

| | ... | PAROLE ELIGIBILITY DATE | EARLIEST POSSIBLE RELEASE DATE | \ |
|---|-----|-------------------------|--------------------------------|-----|
| 0 | ... | 1952-06-20 00:00:00 | | NaN |
| 1 | ... | 1959-06-02 00:00:00 | | LFE |
| 2 | ... | 1972-12-15 00:00:00 | | LFE |
| 3 | ... | | NaN | NaN |
| 4 | ... | | LFE | LFE |

| | GOOD TIME LAW INST RELEASE DATE \ | |
|---|-----------------------------------|------------|
| 0 | 2926 | 1952-08-31 |
| 1 | 2926 | 1967-07-15 |
| 2 | 2926 | 2008-11-25 |
| 3 | 2926 | NaT |
| 4 | 2926 | NaT |

| | INST RELEASE TYPE PAROLE BOARD NEXT REVIEW DATE(MONTH&YEAR) \ | |
|---|---|---------------------|
| 0 | ESCAPE | NaN |
| 1 | ESCAPE | 1959-12-01 00:00:00 |
| 2 | DISCRETIONARY PAROLE | NaN |
| 3 | NaN | 2023-05-01 00:00:00 |
| 4 | NaN | 2024-03-01 00:00:00 |

| | PAROLE BOARD FINAL HEARING DATE(MONTH&YEAR) | PAROLE BOARD STATUS \ |
|---|---|-----------------------|
| 0 | NaT | NaN |
| 1 | NaT | INITIAL REVIEW |
| 2 | NaT | CONTINUED ON PAROLE |
| 3 | NaT | DEFERRED |
| 4 | NaT | DEFERRED |

| | Unnamed: 30 | Unnamed: 31 |
|---|-------------|-------------|
| 0 | NaN | |
| 1 | NaN | |
| 2 | NaN | |
| 3 | NaN | NaN |
| 4 | NaN | NaN |

[5 rows x 32 columns]

```
[9]: df_Active.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 7453 entries, 0 to 7452
```

```
Data columns (total 32 columns):
```

| # | Column | Non-Null Count | Dtype |
|---|---------------------|----------------|----------------|
| 0 | ID NUMBER | 7453 non-null | int64 |
| 1 | COMMITTED LAST NAME | 7453 non-null | object |
| 2 | FIRST NAME | 7453 non-null | object |
| 3 | MIDDLE NAME | 5464 non-null | object |
| 4 | NAME EXTENSION | 7453 non-null | object |
| 5 | LEGAL LAST NAME | 102 non-null | object |
| 6 | FIRST NAME.1 | 102 non-null | object |
| 7 | MIDDLE NAME.1 | 102 non-null | object |
| 8 | NAME EXTENSION.1 | 102 non-null | object |
| 9 | DATE OF BIRTH | 7453 non-null | datetime64[ns] |

```

10 RACE DESC 7453 non-null object
11 GENDER 7453 non-null object
12 FACILITY 7401 non-null object
13 CURRENT SENTENCE PARDONED OR COMMUTED DATE 7453 non-null object
14 GUN CLAUSE 0 non-null float64
15 SENTENCE BEGIN DATE 7450 non-null datetime64[ns]
16 MIN TERM/YEAR 7453 non-null object
17 MIN MONTH 7176 non-null float64
18 MIN DAY 7176 non-null float64
19 MAX TERM/YEAR 7453 non-null object
20 MAX MONTH 7041 non-null float64
21 MAX DAY 7041 non-null float64
22 PAROLE ELIGIBILITY DATE 5444 non-null object
23 EARLIEST POSSIBLE RELEASE DATE 7047 non-null object
24 GOOD TIME LAW 7453 non-null object
25 INST RELEASE DATE 2594 non-null datetime64[ns]
26 INST RELEASE TYPE 2594 non-null object
27 PAROLE BOARD NEXT REVIEW DATE(MONTH&YEAR) 4920 non-null object
28 PAROLE BOARD FINAL HEARING DATE(MONTH&YEAR) 1339 non-null datetime64[ns]
29 PAROLE BOARD STATUS 7449 non-null object
30 Unnamed: 30 0 non-null float64
31 Unnamed: 31 1545 non-null object
dtypes: datetime64[ns](4), float64(6), int64(1), object(21)
memory usage: 1.8+ MB

```

0.5.2 4.2 Assigning a new ‘Active’ Inmate Column

The pandas DataFrame `assign()` method returns a new column, ‘ACTIVE’, that reflects if a prisoner is Active or no longer incarcerated. It uses the DataFrame `.isin` method to check if the full database’s inmates’ ID NUMBER is in the Active inmate database. The DataFrame `.astype` method changes the column to the integer datatype. A new dataframe called ‘df1’ was also created to store this all the previous columns as well as the new Active Column.

```
[10]: df1 = df1.assign(ACTIVE=df1['ID NUMBER']
                      .isin(df_Active['ID NUMBER'])
                      .astype(int))
```

The new column ‘ACTIVE’ is seen at the as last column. If a row has a 1, that means the prisoner is active, and a 0 means they are no longer incarcerated.

```
[11]: df1.head()
```

```
[11]:  ID NUMBER  COMMITTED LAST NAME  FIRST NAME  MIDDLE NAME  NAME EXTENSION  \
0      1702             CLIFFORD    BRADLEY             NaN
1      6145              KANE      THOMAS             NaN
2      6452             ATKINS     LARRY             NaN
3     12444        SHANEYFELT    CHARLEY             NaN
4     15379             BEADES      JOE             NaN
```

| | LEGAL LAST NAME | FIRST NAME | MIDDLE NAME | NAME EXTENSION | DATE OF BIRTH | ... | \ |
|---|-----------------|------------|-------------|----------------|---------------|-----|-----|
| 0 | NaN | NaN | NaN | NaN | NaN | NaN | ... |
| 1 | NaN | NaN | NaN | NaN | 1928-12-21 | NaN | ... |
| 2 | NaN | NaN | NaN | NaN | 1929-07-26 | NaN | ... |
| 3 | NaN | NaN | NaN | NaN | 1905-04-10 | NaN | ... |
| 4 | NaN | NaN | NaN | NaN | 1924-10-12 | NaN | ... |

| | EARLIEST POSSIBLE RELEASE DATE | GOOD TIME LAW | \ |
|---|--------------------------------|---------------|---|
| 0 | NaN | | |
| 1 | NaN | 2926 | |
| 2 | NaN | | |
| 3 | NaN | | |
| 4 | LFE | 2926 | |

| | INST RELEASE DATE | INST RELEASE TYPE | \ |
|---|-------------------|----------------------|---|
| 0 | 1986-01-06 | MANDATORY DISCHARGE | |
| 1 | 1952-08-31 | ESCAPE | |
| 2 | 1955-07-20 | DISCRETIONARY PAROLE | |
| 3 | 1987-12-24 | MANDATORY DISCHARGE | |
| 4 | 1989-07-19 | DISCRETIONARY PAROLE | |

| | PAROLE BOARD NEXT REVIEW DATE(MONTH&YEAR) | \ |
|---|---|---|
| 0 | NaN | |
| 1 | NaN | |
| 2 | NaN | |
| 3 | NaN | |
| 4 | NaN | |

| | PAROLE BOARD FINAL HEARING DATE(MONTH&YEAR) | PAROLE BOARD STATUS | \ |
|---|---|---------------------|---|
| 0 | NaN | NaN | |
| 1 | NaN | NaN | |
| 2 | NaN | PAROLED | |
| 3 | NaN | NaN | |
| 4 | NaN | PAROLED | |

| | PAROLE DATE | PAROLE DISCHARGE | DESC | ACTIVE |
|---|-------------|-------------------------------|------|--------|
| 0 | NaN | NaN | | 0 |
| 1 | NaN | | | 1 |
| 2 | 1980-12-09 | EARLY DISCHARGE BY PAROLE BRD | | 0 |
| 3 | NaN | | | 0 |
| 4 | 1993-01-17 | EARLY DISCHARGE BY PAROLE BRD | | 0 |

[5 rows x 33 columns]

0.6 5. Combining Offense Information with Offense Arrest Group Dataset

A different database contains offense groups and categories for each offense arrest code, values that is found on in df2.

0.6.1 5.1 Examining the Offense Arrest Code Database

The Offense Arrest Code database will be loaded and examined in the same manner as with the full inmate database.

```
[12]: xls = pd.ExcelFile('Offense Arrest Groups.xlsx')
      df_Offense = pd.read_excel(xls, 'Offense Arrest Groups')
```

```
[13]: df_Offense.head()
```

```
[13]:  Offense Arrest CD      Offense Group Offense Category
      0              NaN  Data Unavailable              NaN
      1             A01              Other              Other
      2             A02              Other              Other
      3             A03              Other              Other
      4             A04              Other              Other
```

0.6.2 5.2 Renaming Columns to Match Style

The Columns in this new DataFrame are renamed using the DataFrame rename() method.

```
[14]: df_Offense = df_Offense.rename(columns={
      "Offense Arrest CD": "OFFENSE ARREST CD", "Offense Arrest":
      "OFFENSE ARREST", "Offense Group": "OFFENSE GROUP",
      "Offense Category": "OFFENSE CATEGORY"})
```

```
[15]: df2 = df2.rename(columns={
      "Offense Arrest CD": "OFFENSE ARREST CD", "Offense Arrest":
      "OFFENSE ARREST", "Offense Group": "OFFENSE GROUP"
      })
```

0.6.3 5.3 Merging the DataFrame with the Offense Arrest Code Database

The DataFrame merge() method will combine dataframes on a specific row with various specifications.

```
[16]: df2 = (df2.merge(df_Offense, on='OFFENSE ARREST CD', how='left'))
```

The new columns can be seen.

```
[17]: df2.head()
```

```
[17]:  ID  ID NUMBER OFFENSE MINIMUM YEAR OR TERM  MINIMUM MONTH  MINIMUM DAY  \
      0   1      6145              1          0.0          0.0
      1   2      6452              2          0.0          0.0
```

| | | | | | |
|---|---|-------|----|-----|-----|
| 2 | 3 | 12444 | 1 | 0.0 | 0.0 |
| 3 | 4 | 15379 | 10 | 0.0 | 0.0 |
| 4 | 5 | 15410 | 2 | 0.0 | 0.0 |

| | OFFENSE | MAXIMUM YEAR OR TERM | MAXIMUM MONTH | MAXIMUM DAY | \ |
|---|---------|----------------------|---------------|-------------|---|
| 0 | | 3 | 0.0 | 0.0 | |
| 1 | | 10 | 0.0 | 0.0 | |
| 2 | | 9 | 0.0 | 0.0 | |
| 3 | | LFE | NaN | NaN | |
| 4 | | 3 | 0.0 | 0.0 | |

| | OFFENSE | ARREST DESC | FELONY MSDMNR CODE | OFFENSE TYPE CODE | \ |
|---|---------|--------------------|--------------------|-------------------|---|
| 0 | | CATTLE STEALING | FELONY | * | |
| 1 | | FORGERY 2ND DEGREE | NaN | * | |
| 2 | | BURGLARY | FELONY | * | |
| 3 | | MURDER 2ND DEGREE | NaN | * | |
| 4 | | FORGERY 1ST DEGREE | FELONY | * | |

| | OFFENSE | ATTEMPT DESC | HABITUAL CRIMINAL | OFFENSE RUN CODE | COUNTY COMMITTED | \ |
|---|---------|--------------|-------------------|------------------|------------------|---|
| 0 | | NaN | NaN | CC | MADISON | |
| 1 | | NaN | NaN | CC | SCOTTS BLUFF | |
| 2 | | NaN | NaN | CC | HAMILTON | |
| 3 | | NaN | NaN | CC | DOUGLAS | |
| 4 | | NaN | NaN | CC | SCOTTS BLUFF | |

| | OFFENSE | ARREST CD | OFFENSE | ARREST | OFFENSE GROUP | OFFENSE CATEGORY |
|---|---------|-----------|--------------------|----------|---------------|------------------|
| 0 | | D21 | | THEFT | Theft | Property |
| 1 | | E02 | FORGERY 2ND DEGREE | | Fraud | Other |
| 2 | | D11 | | BURGLARY | Burglary | Property |
| 3 | | B02 | MURDER 2ND DEGREE | | Homicide | Person |
| 4 | | E01 | FORGERY 1ST DEGREE | | Fraud | Other |

0.7 6. Initial Data Cleaning

Datasets are almost always imperfect and this can hinder future analysis.

0.7.1 6.1 Dropping unneeded columns

The columns that were removed contained unnessescary personal information about the inmates or their sentences that were not of use in this research's more macro-based lense. Parole information was also referenced by the NDCS to be fairly incomplete and too difficult to research.

Unneeded columns can be deleted using the DataFrame drop method.

```
[18]: df1 = df1.drop(['FIRST NAME',
                    'FIRST NAME',
                    'MIDDLE NAME',
                    'MIDDLE NAME',
```

```

'NAME EXTENSION',
'COMMITTED LAST NAME',
'LEGAL LAST NAME',
'NAME EXTENSION',
'FIRST NAME2',
'MIDDLE NAME3',
'NAME EXTENSION4',
'GUN CLAUSE',
'MIN MONTH',
'CURRENT SENTENCE PARDONED OR COMMUTED DATE',
'MIN DAY',
'MAX MONTH',
'MAX DAY',
'PAROLE ELIGIBILITY DATE',
'GOOD TIME LAW',
'EARLIEST POSSIBLE RELEASE DATE',
'INST RELEASE TYPE',
'PAROLE BOARD NEXT REVIEW DATE(MONTH&YEAR)',
'PAROLE BOARD FINAL HEARING DATE(MONTH&YEAR)',
'PAROLE BOARD STATUS',
'PAROLE DISCHARGE DESC',
'PAROLE DATE'],axis=1)

```

```

[19]: df2 = df2.drop(['MINIMUM MONTH',
                    'ID',
                    'MINIMUM DAY',
                    'MAXIMUM MONTH',
                    'MAXIMUM DAY'],axis=1)

```

```

[20]: df1.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72954 entries, 0 to 72953
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID NUMBER             72954 non-null  int64
1   DATE OF BIRTH         72940 non-null  datetime64[ns]
2   RACE DESC             72954 non-null  object
3   GENDER                72954 non-null  object
4   FACILITY              14080 non-null  object
5   SENTENCE BEGIN DATE   71475 non-null  datetime64[ns]
6   MIN TERM/YEAR         72954 non-null  object
7   MAX TERM/YEAR         72954 non-null  object
8   INST RELEASE DATE     67994 non-null  datetime64[ns]
9   ACTIVE                72954 non-null  int64
dtypes: datetime64[ns](3), int64(2), object(5)

```

memory usage: 5.6+ MB

```
[21]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129897 entries, 0 to 129896
Data columns (total 14 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   ID NUMBER                            129897 non-null int64
 1   OFFENSE MINIMUM YEAR OR TERM        129897 non-null object
 2   OFFENSE MAXIMUM YEAR OR TERM        129897 non-null object
 3   OFFENSE ARREST DESC                 129897 non-null object
 4   FELONY MSDMNR CODE                  115048 non-null object
 5   OFFENSE TYPE CODE                   129897 non-null object
 6   OFFENSE ATTEMPT DESC                16430 non-null object
 7   HABITUAL CRIMINAL                   767 non-null   object
 8   OFFENSE RUN CODE                    105476 non-null object
 9   COUNTY COMMITTED                    129869 non-null object
10   OFFENSE ARREST CD                   129826 non-null object
11   OFFENSE ARREST                      129826 non-null object
12   OFFENSE GROUP                       129897 non-null object
13   OFFENSE CATEGORY                    129826 non-null object
dtypes: int64(1), object(13)
memory usage: 14.9+ MB
```

```
[22]: df2.head()
```

```
[22]:   ID NUMBER  OFFENSE MINIMUM YEAR OR TERM  OFFENSE MAXIMUM YEAR OR TERM  \
0         6145                             1                             3
1         6452                             2                             10
2        12444                             1                             9
3        15379                             10                          LFE
4        15410                             2                             3

   OFFENSE ARREST DESC  FELONY MSDMNR CODE  OFFENSE TYPE CODE  \
0      CATTLE STEALING          FELONY          *
1    FORGERY 2ND DEGREE           NaN          *
2           BURGLARY          FELONY          *
3    MURDER 2ND DEGREE           NaN          *
4    FORGERY 1ST DEGREE          FELONY          *

   OFFENSE ATTEMPT DESC  HABITUAL CRIMINAL  OFFENSE RUN CODE  COUNTY COMMITTED  \
0                NaN                NaN                CC          MADISON
1                NaN                NaN                CC    SCOTTS BLUFF
2                NaN                NaN                CC          HAMILTON
3                NaN                NaN                CC          DOUGLAS
```

| | | | | | | | |
|---|--|-----|--|-----|--|----|--------------|
| 4 | | NaN | | NaN | | CC | SCOTTS BLUFF |
|---|--|-----|--|-----|--|----|--------------|

| | OFFENSE | ARREST CD | OFFENSE | ARREST | OFFENSE | GROUP | OFFENSE | CATEGORY |
|---|---------|-----------|---------|------------|---------|----------|---------|----------|
| 0 | | D21 | | THEFT | | Theft | | Property |
| 1 | | E02 | FORGERY | 2ND DEGREE | | Fraud | | Other |
| 2 | | D11 | | BURGLARY | | Burglary | | Property |
| 3 | | B02 | MURDER | 2ND DEGREE | | Homicide | | Person |
| 4 | | E01 | FORGERY | 1ST DEGREE | | Fraud | | Other |

0.7.2 6.2 Merging DataFrame 1 and DataFrame 2

These two pages are combined to focus on the offense data instead of the individual inmates'.

```
[23]: df3 = (df1.merge(df2, on='ID NUMBER'))
```

```
[24]: df3.head()
```

```
[24]:
```

| | ID NUMBER | DATE OF BIRTH | | RACE | DESC | GENDER | \ |
|---|-----------|---------------|-------|------|------|--------|---|
| 0 | 6145 | 1928-12-21 | WHITE | | | MALE | |
| 1 | 6452 | 1929-07-26 | WHITE | | | MALE | |
| 2 | 12444 | 1905-04-10 | WHITE | | | MALE | |
| 3 | 15379 | 1924-10-12 | WHITE | | | MALE | |
| 4 | 15410 | NaT | WHITE | | | MALE | |

| | FACILITY | SENTENCE | BEGIN DATE | MIN | TERM/YEAR | \ |
|---|-----------------------------|----------|------------|-----|-----------|----|
| 0 | NEBRASKA STATE PENITENTIARY | | 1952-06-20 | | | 1 |
| 1 | NEBRASKA STATE PENITENTIARY | | 1953-11-25 | | | 2 |
| 2 | | NaN | 1935-10-15 | | | 1 |
| 3 | | NaN | 1945-05-02 | | | 10 |
| 4 | NEBRASKA STATE PENITENTIARY | | 1945-05-28 | | | 2 |

| | MAX TERM/YEAR | INST | RELEASE DATE | ACTIVE | ... | FELONY | MSDMNR | CODE | \ |
|---|---------------|------|--------------|--------|-----|--------|--------|--------|---|
| 0 | 3 | | 1952-08-31 | 1 | ... | | | FELONY | |
| 1 | 10 | | 1955-07-20 | 0 | ... | | | NaN | |
| 2 | 9 | | 1987-12-24 | 0 | ... | | | FELONY | |
| 3 | LFE | | 1989-07-19 | 0 | ... | | | NaN | |
| 4 | 3 | | 2003-03-13 | 0 | ... | | | FELONY | |

| | OFFENSE | TYPE | CODE | OFFENSE | ATTEMPT | DESC | HABITUAL | CRIMINAL | OFFENSE | RUN | CODE | \ |
|---|---------|------|------|---------|---------|------|----------|----------|---------|-----|------|---|
| 0 | | * | | | | NaN | | NaN | | | CC | |
| 1 | | * | | | | NaN | | NaN | | | CC | |
| 2 | | * | | | | NaN | | NaN | | | CC | |
| 3 | | * | | | | NaN | | NaN | | | CC | |
| 4 | | * | | | | NaN | | NaN | | | CC | |

| | COUNTY | COMMITTED | OFFENSE | ARREST CD | OFFENSE | ARREST | OFFENSE | GROUP | \ |
|---|---------|-----------|---------|-----------|---------|--------|---------|-------|---|
| 0 | MADISON | | | D21 | | THEFT | | Theft | |

| | | | | |
|---|--------------|-----|--------------------|----------|
| 1 | SCOTTS BLUFF | E02 | FORGERY 2ND DEGREE | Fraud |
| 2 | HAMILTON | D11 | BURGLARY | Burglary |
| 3 | DOUGLAS | B02 | MURDER 2ND DEGREE | Homicide |
| 4 | SCOTTS BLUFF | E01 | FORGERY 1ST DEGREE | Fraud |

| | OFFENSE CATEGORY |
|---|------------------|
| 0 | Property |
| 1 | Other |
| 2 | Property |
| 3 | Person |
| 4 | Other |

[5 rows x 23 columns]

```
[25]: df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129897 entries, 0 to 129896
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                            129897 non-null int64
1   DATE OF BIRTH                        129885 non-null datetime64[ns]
2   RACE DESC                            129897 non-null object
3   GENDER                              129897 non-null object
4   FACILITY                             29995 non-null object
5   SENTENCE BEGIN DATE                 127994 non-null datetime64[ns]
6   MIN TERM/YEAR                       129897 non-null object
7   MAX TERM/YEAR                       129897 non-null object
8   INST RELEASE DATE                   117740 non-null datetime64[ns]
9   ACTIVE                              129897 non-null int64
10  OFFENSE MINIMUM YEAR OR TERM         129897 non-null object
11  OFFENSE MAXIMUM YEAR OR TERM         129897 non-null object
12  OFFENSE ARREST DESC                  129897 non-null object
13  FELONY MSDMNR CODE                   115048 non-null object
14  OFFENSE TYPE CODE                    129897 non-null object
15  OFFENSE ATTEMPT DESC                  16430 non-null object
16  HABITUAL CRIMINAL                    767 non-null object
17  OFFENSE RUN CODE                     105476 non-null object
18  COUNTY COMMITTED                     129869 non-null object
19  OFFENSE ARREST CD                    129826 non-null object
20  OFFENSE ARREST                       129826 non-null object
21  OFFENSE GROUP                        129897 non-null object
22  OFFENSE CATEGORY                      129826 non-null object
dtypes: datetime64[ns](3), int64(2), object(18)
memory usage: 23.8+ MB
```

0.7.3 6.3 Checking to see if All Inmate Records were transferred properly

We can check to see if an inmate that is serving for multiple offenses is properly accounted for.

```
[26]: df3.loc[df3['ID NUMBER'] == 61210]
```

```
[26]:
```

| | ID NUMBER | DATE OF BIRTH | RACE | DESC | GENDER | \ |
|-------|-----------|---------------|-------|------|--------|---|
| 52677 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52678 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52679 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52680 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52681 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52682 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52683 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52684 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52685 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52686 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52687 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52688 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52689 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52690 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52691 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52692 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52693 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52694 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52695 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52696 | 61210 | 1970-02-19 | BLACK | | MALE | |
| 52697 | 61210 | 1970-02-19 | BLACK | | MALE | |

| | FACILITY | SENTENCE | BEGIN DATE | MIN | TERM/YEAR | \ |
|-------|----------------|--------------|------------|-----|-----------|---|
| 52677 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52678 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52679 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52680 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52681 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52682 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52683 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52684 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52685 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52686 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52687 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52688 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52689 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52690 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52691 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52692 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52693 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |
| 52694 | NEBRASKA STATE | PENITENTIARY | 2020-01-22 | | 0 | |

| | | | |
|-------|-----------------------------|------------|---|
| 52695 | NEBRASKA STATE PENITENTIARY | 2020-01-22 | 0 |
| 52696 | NEBRASKA STATE PENITENTIARY | 2020-01-22 | 0 |
| 52697 | NEBRASKA STATE PENITENTIARY | 2020-01-22 | 0 |

| | MAX TERM/YEAR | INST RELEASE DATE | ACTIVE | ... | FELONY MSDMNR CODE | \ |
|-------|---------------|-------------------|--------|-----|--------------------|---|
| 52677 | 0 | 2020-10-14 | 1 | ... | CLASS III FELONY | |
| 52678 | 0 | 2020-10-14 | 1 | ... | CLASS III FELONY | |
| 52679 | 0 | 2020-10-14 | 1 | ... | CLASS III FELONY | |
| 52680 | 0 | 2020-10-14 | 1 | ... | CLASS III FELONY | |
| 52681 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52682 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52683 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52684 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52685 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52686 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52687 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52688 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52689 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52690 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52691 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52692 | 0 | 2020-10-14 | 1 | ... | CLASS III FELONY | |
| 52693 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52694 | 0 | 2020-10-14 | 1 | ... | CLASS II FELONY | |
| 52695 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52696 | 0 | 2020-10-14 | 1 | ... | CLASS IV FELONY | |
| 52697 | 0 | 2020-10-14 | 1 | ... | CLASS IIIA FELONY | |

| | OFFENSE TYPE CODE | OFFENSE ATTEMPT | DESC | HABITUAL | CRIMINAL | \ |
|-------|-------------------|-----------------|------|----------|----------|---|
| 52677 | * | | NaN | | NaN | |
| 52678 | * | | NaN | | NaN | |
| 52679 | * | | NaN | | NaN | |
| 52680 | * | | NaN | | NaN | |
| 52681 | A | | NaN | | NaN | |
| 52682 | A | | NaN | | NaN | |
| 52683 | A | | NaN | | NaN | |
| 52684 | A | | NaN | | NaN | |
| 52685 | A | | NaN | | NaN | |
| 52686 | A | | NaN | | NaN | |
| 52687 | A | | NaN | | NaN | |
| 52688 | A | | NaN | | NaN | |
| 52689 | A | | NaN | | NaN | |
| 52690 | A | | NaN | | NaN | |
| 52691 | A | | NaN | | NaN | |
| 52692 | A | CONSPIRACY | | | NaN | |
| 52693 | B | | NaN | | NaN | |
| 52694 | B | | NaN | | NaN | |
| 52695 | C | | NaN | | NaN | |

| | | | |
|-------|---|-----|-----|
| 52696 | D | NaN | NaN |
| 52697 | D | NaN | NaN |

| | OFFENSE RUN CODE | COUNTY COMMITTED | OFFENSE ARREST CD \ |
|-------|------------------|------------------|---------------------|
| 52677 | NaN | DOUGLAS | E02 |
| 52678 | CC | DOUGLAS | E02 |
| 52679 | CC | DOUGLAS | E02 |
| 52680 | CC | DOUGLAS | E02 |
| 52681 | CC | DOUGLAS | E02 |
| 52682 | CC | DOUGLAS | E02 |
| 52683 | CC | DOUGLAS | E02 |
| 52684 | CC | DOUGLAS | E02 |
| 52685 | CC | DOUGLAS | E02 |
| 52686 | CC | DOUGLAS | E02 |
| 52687 | CC | DOUGLAS | E02 |
| 52688 | CC | DOUGLAS | E02 |
| 52689 | CC | DOUGLAS | E02 |
| 52690 | CC | DOUGLAS | E02 |
| 52691 | CC | DOUGLAS | E02 |
| 52692 | CS | DOUGLAS | D45 |
| 52693 | CC | DOUGLAS | M02 |
| 52694 | CS | DOUGLAS | M01 |
| 52695 | CS | DOUGLAS | M02 |
| 52696 | CC | DOUGLAS | M02 |
| 52697 | CC | DOUGLAS | M03 |

| | OFFENSE ARREST | OFFENSE GROUP | OFFENSE CATEGORY |
|-------|-------------------------------|---------------|------------------|
| 52677 | FORGERY 2ND DEGREE | Fraud | Other |
| 52678 | FORGERY 2ND DEGREE | Fraud | Other |
| 52679 | FORGERY 2ND DEGREE | Fraud | Other |
| 52680 | FORGERY 2ND DEGREE | Fraud | Other |
| 52681 | FORGERY 2ND DEGREE | Fraud | Other |
| 52682 | FORGERY 2ND DEGREE | Fraud | Other |
| 52683 | FORGERY 2ND DEGREE | Fraud | Other |
| 52684 | FORGERY 2ND DEGREE | Fraud | Other |
| 52685 | FORGERY 2ND DEGREE | Fraud | Other |
| 52686 | FORGERY 2ND DEGREE | Fraud | Other |
| 52687 | FORGERY 2ND DEGREE | Fraud | Other |
| 52688 | FORGERY 2ND DEGREE | Fraud | Other |
| 52689 | FORGERY 2ND DEGREE | Fraud | Other |
| 52690 | FORGERY 2ND DEGREE | Fraud | Other |
| 52691 | FORGERY 2ND DEGREE | Fraud | Other |
| 52692 | THEFT BY DECEPTION | Theft | Property |
| 52693 | DRIVING UNDER REVOKED LICENSE | Motor Vehicle | Other |
| 52694 | DRIVING WHILE INTOXICATED | Motor Vehicle | Other |
| 52695 | DRIVING UNDER REVOKED LICENSE | Motor Vehicle | Other |
| 52696 | DRIVING UNDER REVOKED LICENSE | Motor Vehicle | Other |

```
52697  DRIVING UNDER INFLUENCE/INJURY  Motor Vehicle      Person
```

```
[21 rows x 23 columns]
```

0.7.4 6.4 Dropping All “NA” Missing Features

We can check if any columns have missing data values and count them by using the `isnull()` method and `sum()` method.

```
[27]: df3['SENTENCE BEGIN DATE'].isnull().sum()
```

```
[27]: 1903
```

Missing data values or (NA) is removed from the data in certain columns. The `DataFrame.dropna()` method can be used to do this.

We drop all inmates who have an unknown sentence begin date.

```
[28]: df3 = df3.dropna(axis=0, how="any", subset=['SENTENCE BEGIN DATE'])
```

We can check to see if this function worked.

```
[29]: df3['SENTENCE BEGIN DATE'].isnull().sum()
```

```
[29]: 0
```

We drop all inmates who have an unknown date of birth.

```
[30]: df3['DATE OF BIRTH'].isnull().sum()
```

```
[30]: 5
```

```
[31]: df3 = df3.dropna(axis=0, how="any", subset=['DATE OF BIRTH'])
```

0.7.5 6.5 Checking for Duplicate Rows

Sometimes data contains duplicate rows of information that may skew future analysis. These can be found by using the `DataFrame.duplicated()` method.

```
[32]: duplicate = df3[df3.duplicated()]
duplicate
```

```
[32]:
```

| | ID NUMBER | DATE OF BIRTH | RACE DESC | GENDER | \ |
|-----|-----------|---------------|-----------|--------|---|
| 7 | 16657 | 1929-01-10 | WHITE | MALE | |
| 16 | 19595 | 1933-01-28 | HISPANIC | MALE | |
| 24 | 20560 | 1941-01-02 | WHITE | MALE | |
| 30 | 20841 | 1942-08-28 | WHITE | MALE | |
| 94 | 25594 | 1934-10-21 | WHITE | MALE | |
| ... | ... | ... | ... | ... | |

| | | | | |
|--------|--------|------------|-------|--------|
| 129640 | 502039 | 1987-06-04 | WHITE | MALE |
| 129646 | 502041 | 1989-10-16 | WHITE | FEMALE |
| 129702 | 502083 | 1994-03-28 | BLACK | MALE |
| 129725 | 502098 | 1994-07-24 | OTHER | MALE |
| 129738 | 502108 | 1987-02-07 | WHITE | MALE |

| | FACILITY | SENTENCE | BEGIN DATE | MIN TERM/YEAR | \ |
|--------|-----------------------------|----------|------------|---------------|---|
| 7 | | NaN | 1948-12-22 | LFE | |
| 16 | NEBRASKA STATE PENITENTIARY | | 1956-05-16 | 3 | |
| 24 | NEBRASKA STATE PENITENTIARY | | 1958-10-14 | 25 | |
| 30 | NEBRASKA STATE PENITENTIARY | | 1959-06-02 | 10 | |
| 94 | NEBRASKA STATE PENITENTIARY | | 1967-04-10 | 3 | |
| ... | ... | | ... | ... | |
| 129640 | | NaN | 2011-03-23 | 0 | |
| 129646 | | NaN | 2011-03-28 | 0 | |
| 129702 | | NaN | 2011-07-28 | 0 | |
| 129725 | | NaN | 2011-09-21 | 0 | |
| 129738 | | NaN | 2011-10-07 | 0 | |

| | MAX TERM/YEAR | INST | RELEASE DATE | ACTIVE | ... | FELONY MSDMNR CODE | \ |
|--------|---------------|------|--------------|--------|-----|---------------------|---|
| 7 | LFE | | 2002-12-27 | 0 | ... | FELONY | |
| 16 | 15 | | 1961-03-22 | 0 | ... | NaN | |
| 24 | 45 | | 1974-06-21 | 0 | ... | NaN | |
| 30 | LFE | | 1967-07-15 | 1 | ... | FELONY | |
| 94 | 15 | | 1975-09-30 | 0 | ... | NaN | |
| ... | ... | | ... | ... | | ... | |
| 129640 | 0 | | 2011-08-21 | 0 | ... | CLASS III FELONY | |
| 129646 | 0 | | 2011-09-01 | 0 | ... | CLASS III FELONY | |
| 129702 | 0 | | 2011-09-01 | 0 | ... | CLASS I MISDEMEANOR | |
| 129725 | 0 | | 2012-01-26 | 0 | ... | CLASS IV FELONY | |
| 129738 | 0 | | 2012-03-11 | 0 | ... | CLASS IV FELONY | |

| | OFFENSE TYPE | CODE | OFFENSE ATTEMPT | DESC | HABITUAL | CRIMINAL | \ |
|--------|--------------|------|-----------------|------|----------|----------|---|
| 7 | | * | | NaN | | NaN | |
| 16 | | * | | NaN | | NaN | |
| 24 | | * | | NaN | | NaN | |
| 30 | | * | | NaN | | NaN | |
| 94 | | * | | NaN | | NaN | |
| ... | ... | | ... | | ... | | |
| 129640 | | * | | NaN | | NaN | |
| 129646 | | * | AID & ABET | | | NaN | |
| 129702 | | * | | NaN | | NaN | |
| 129725 | | * | | NaN | | NaN | |
| 129738 | | * | | NaN | | NaN | |

| | OFFENSE RUN | CODE | COUNTY COMMITTED | OFFENSE | ARREST CD | \ |
|---|-------------|------|------------------|---------|-----------|---|
| 7 | | CC | DOUGLAS | | B02 | |

| | | | |
|--------|-----|-----------|-----|
| 16 | CC | KEITH | B11 |
| 24 | CC | BOONE | B02 |
| 30 | CC | DOUGLAS | B02 |
| 94 | CC | GAGE | B41 |
| ... | ... | ... | ... |
| 129640 | NaN | MORRILL | D11 |
| 129646 | NaN | LINCOLN | H50 |
| 129702 | NaN | NEMAHA | M11 |
| 129725 | NaN | LANCASTER | B14 |
| 129738 | NaN | NANCE | D31 |

| | OFFENSE | ARREST | OFFENSE GROUP | OFFENSE CATEGORY |
|--------|---------------------|------------|---------------|------------------|
| 7 | MURDER | 2ND DEGREE | Homicide | Person |
| 16 | ASSAULT | 1ST DEGREE | Assault | Person |
| 24 | MURDER | 2ND DEGREE | Homicide | Person |
| 30 | MURDER | 2ND DEGREE | Homicide | Person |
| 94 | | ROBBERY | Robbery | Person |
| ... | | ... | ... | ... |
| 129640 | | BURGLARY | Burglary | Property |
| 129646 | | ESCAPE | Other | Other |
| 129702 | CLASS I MISDEMEANOR | | Other | Other |
| 129725 | TERRORISTIC THREATS | | Assault | Person |
| 129738 | CRIMINAL MISCHIEF | | Other | Other |

[8884 rows x 23 columns]

The duplicate rows are removed by using DataFrame `drop_duplicates()`.

```
[33]: df3 = df3.drop_duplicates()
```

The overall number of entries will change.

```
[34]: df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119105 entries, 0 to 129896
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                            119105 non-null  int64
1   DATE OF BIRTH                        119105 non-null  datetime64[ns]
2   RACE DESC                            119105 non-null  object
3   GENDER                              119105 non-null  object
4   FACILITY                             27415 non-null   object
5   SENTENCE BEGIN DATE                 119105 non-null  datetime64[ns]
6   MIN TERM/YEAR                       119105 non-null  object
7   MAX TERM/YEAR                       119105 non-null  object
8   INST RELEASE DATE                   107890 non-null  datetime64[ns]
```

```

9    ACTIVE                                119105 non-null  int64
10   OFFENSE MINIMUM YEAR OR TERM         119105 non-null  object
11   OFFENSE MAXIMUM YEAR OR TERM         119105 non-null  object
12   OFFENSE ARREST DESC                  119105 non-null  object
13   FELONY MSDMNR CODE                   105275 non-null  object
14   OFFENSE TYPE CODE                    119105 non-null  object
15   OFFENSE ATTEMPT DESC                  15577 non-null  object
16   HABITUAL CRIMINAL                    706 non-null    object
17   OFFENSE RUN CODE                     96660 non-null  object
18   COUNTY COMMITTED                     119082 non-null  object
19   OFFENSE ARREST CD                    119051 non-null  object
20   OFFENSE ARREST                       119051 non-null  object
21   OFFENSE GROUP                        119105 non-null  object
22   OFFENSE CATEGORY                     119051 non-null  object
dtypes: datetime64[ns](3), int64(2), object(18)
memory usage: 21.8+ MB

```

0.8 7. Adding in Inmate Age Columns for Further Analysis

Some columns will be added using information from the original columns to help expand the analysis.

0.8.1 7.1 Changing Column Data Types to Datetime

Datetime is a data type that is used for dates and times. The Pandas `to_datetime` method changes an object data type to a datetime format.

```
[35]: df3['SENTENCE BEGIN DATE DT'] = pd.to_datetime(df3['SENTENCE BEGIN DATE'])
      df3['DATE OF BIRTH DT'] = pd.to_datetime(df3['DATE OF BIRTH'])
```

0.8.2 7.2 Finding Age of Inmates at Time of Incarceration

This age will be helpful to see at what age inmates were incarcerated and how that may have changed over time

Datetime also is a Python module that can manipulate the datetime datatype. `Timedelta` is a function from datetime that is used to calculate the difference between dates. The Sentence Begin Age can be found through finding the number of days between the sentence begin date and the inmate's date of birth. This number is then divided by 365 days to get their age in years.

```
[36]: import datetime as dt
      from datetime import timedelta

      df3['SENTENCE BEGIN AGE DAYS'] = df3['SENTENCE BEGIN DATE DT'] - df3['DATE OF_
      ↪BIRTH DT']
      df3['SENTENCE BEGIN AGE'] = df3['SENTENCE BEGIN AGE DAYS'] / timedelta(days=365)
```

0.8.3 7.3 Finding Current Age of Inmates

This age will be helpful to see how old inmates are now, especially for active ones.

The 'date' is set to the current local date and time using the Pandas `datetime.now()` method. `.strftime` is the date in a string format that is given in Year/Month/Day. The Current Age can be found through finding the number of days between the current date and the inmate's date of birth. This number is then divided by 365 days to get their age in years.

```
[37]: date = pd.datetime.now().strftime('%Y%m%d')
      date = pd.to_datetime(date,format='%Y%m%d')
      df3['CURRENT AGE DAYS'] = date - df3['DATE OF BIRTH DT']
      df3['CURRENT AGE'] = df3['CURRENT AGE DAYS'] / timedelta(days=365)
```

<ipython-input-37-8ccf1783ddcb>:1: FutureWarning: The pandas.datetime class is deprecated and will be removed from pandas in a future version. Import from datetime module instead.

```
date = pd.datetime.now().strftime('%Y%m%d')
```

We can drop these columns since they were just made temporarily to get the ages in years.

```
[38]: df3 = df3.drop([
      'CURRENT AGE DAYS',
      'SENTENCE BEGIN AGE DAYS'],axis=1)
```

0.8.4 7.4 Making a New Column for the Sentence Begin Year

This column will be helpful to later separate inmates into different decade cohorts for analysis.

The datetime (dt) `.year` function can be used to find just the year portion of the full date.

```
[39]: df3['SENTENCE BEGIN YEAR'] = df3['SENTENCE BEGIN DATE DT'].dt.year
```

0.9 8.0 Altering Life Sentencing Values

Some inmates receive life sentences in their MIN TERM/YEAR or MAX TERM/YEAR, and are given the value 'LFE' instead of an integer. This object value limits evaluation on average MIN/MAX TERM of the inmates. These values are changed to numerical representations of how much life left the inmates have.

0.9.1 8.1 Declaring Average Lifespan Values and Finding Years Between Lifespan and Inmate Sentence Begin Age

lfeM represents the average lifespan of a male Nebraskan, and lfeF represents the average lifespan of a female Nebraskan. These values were obtained from (LINK). The columns LFE SENTENCE M and LFE SENTENCE F are created from taking the respective average lifespans subtracted by the inmates' age at the start of their incarceration.

```
[40]: lfeM = 77.7
      lfeF = 81.89
```

```
df3['LFE SENTENCE M'] = lfeM - df3['SENTENCE BEGIN AGE']
df3['LFE SENTENCE F'] = lfeF - df3['SENTENCE BEGIN AGE']
```

0.9.2 8.2 Assigning Found Values to a Life Sentence Column

The `lfe_sentence()` function looks to see if the row has the a male or female inmate, and returns the corresponding LFE SENTENCE column. A new column, LFE SENTENCE MIN/MAX is created from applying the `lfe_sentence()` function to our data. The values in LFE SENTENCE MIN/MAX are then rounded using `.round()`.

```
[41]: def lfe_sentence(df3):
        if df3['GENDER'] == 'MALE':
            return df3['LFE SENTENCE M']
        else:
            return df3['LFE SENTENCE F']

df3['LFE SENTENCE MIN/MAX'] = df3.apply(lfe_sentence,axis=1)
df3['LFE SENTENCE MIN/MAX'] = df3['LFE SENTENCE MIN/MAX'].round()
```

0.9.3 8.3 Replace OFFENSE MINIMUM or OFFENSE MAXIMUM YEAR OR TERM Values with Life Sentence Age Values

Now the original MIN or MAX TERM/YEAR value of 'LFE' is replaced with the value in LFE SENTENCE MIN/MAX by using a numpy where method. The where method lets you find and replace values.

```
[42]: df3['OFFENSE MINIMUM YEAR OR TERM'] = np.where(df3['OFFENSE MINIMUM YEAR OR_
↳TERM'] == 'LFE',
                                                    df3['LFE SENTENCE MIN/MAX'], df3['OFFENSE_
↳MINIMUM YEAR OR TERM'])
df3['OFFENSE MAXIMUM YEAR OR TERM'] = np.where(df3['OFFENSE MAXIMUM YEAR OR_
↳TERM'] == 'LFE',
                                                    df3['LFE SENTENCE MIN/MAX'], df3['OFFENSE_
↳MAXIMUM YEAR OR TERM'])
```

0.9.4 8.4 Dropping Old Columns

The columns used to create these life sentence replacement values can now be deleted.

```
[43]: df3=df3.drop([
        'LFE SENTENCE M',
        'LFE SENTENCE F',
        'LFE SENTENCE MIN/MAX'],axis=1)
```

```
[44]: df3.head()
```

```

[44]: ID NUMBER DATE OF BIRTH RACE DESC GENDER \
0      6145      1928-12-21 WHITE MALE
1      6452      1929-07-26 WHITE MALE
2     12444      1905-04-10 WHITE MALE
3     15379      1924-10-12 WHITE MALE
6     16657      1929-01-10 WHITE MALE

FACILITY SENTENCE BEGIN DATE MIN TERM/YEAR \
0 NEBRASKA STATE PENITENTIARY 1952-06-20 1
1 NEBRASKA STATE PENITENTIARY 1953-11-25 2
2 NaN 1935-10-15 1
3 NaN 1945-05-02 10
6 NaN 1948-12-22 LFE

MAX TERM/YEAR INST RELEASE DATE ACTIVE ... COUNTY COMMITTED \
0      3      1952-08-31      1 ... MADISON
1     10      1955-07-20      0 ... SCOTTS BLUFF
2      9      1987-12-24      0 ... HAMILTON
3     LFE      1989-07-19      0 ... DOUGLAS
6     LFE      2002-12-27      0 ... DOUGLAS

OFFENSE ARREST CD OFFENSE ARREST OFFENSE GROUP OFFENSE CATEGORY \
0      D21 THEFT Theft Property
1      E02 FORGERY 2ND DEGREE Fraud Other
2      D11 BURGLARY Burglary Property
3      B02 MURDER 2ND DEGREE Homicide Person
6      B02 MURDER 2ND DEGREE Homicide Person

SENTENCE BEGIN DATE DT DATE OF BIRTH DT SENTENCE BEGIN AGE CURRENT AGE \
0      1952-06-20 1928-12-21 23.512329 92.273973
1      1953-11-25 1929-07-26 24.350685 91.679452
2      1935-10-15 1905-04-10 30.534247 115.989041
3      1945-05-02 1924-10-12 20.567123 96.468493
6      1948-12-22 1929-01-10 19.961644 92.219178

SENTENCE BEGIN YEAR
0      1952
1      1953
2      1935
3      1945
6      1948

```

[5 rows x 28 columns]

0.10 9.0 Deleting Death and Independent Values

Very few of the MIN or MAX TERM/YEAR values are labelled with DTH or IND. These stand for Death and Independent sentences, and only hinder analysis on average sentencing. They are removed to simplify the data, as they account for fewer than 30 inmates. These few inmates act as major outliers compared to the rest of the database.

0.10.1 9.1 Checking to See if IND or DTH Values Exist

The DataFrame method `str.contains()` finds if the given value is in the DataFrame being analyzed. `.any()` returns a true or false statement if the given value is found in the DataFrame.

```
[45]: df3['OFFENSE MINIMUM YEAR OR TERM'].str.contains('IND').any()
```

```
[45]: True
```

```
[46]: df3['OFFENSE MINIMUM YEAR OR TERM'].str.contains('DTH').any()
```

```
[46]: True
```

0.10.2 9.2 Drop rows where values equal IND or DTH

The `drop()` method is used again to delete any columns where the MIN or MAX TERM/YEAR value is equal to DTH or IND.

```
[47]: df3.drop(df3.loc[df3['OFFENSE MINIMUM YEAR OR TERM']=='DTH'].index,
            inplace=True)
df3.drop(df3.loc[df3['OFFENSE MINIMUM YEAR OR TERM']=='IND'].index,
            inplace=True)

df3.drop(df3.loc[df3['OFFENSE MINIMUM YEAR OR TERM']=='DTH'].index,
            inplace=True)
df3.drop(df3.loc[df3['OFFENSE MAXIMUM YEAR OR TERM']=='IND'].index,
            inplace=True)
```

0.10.3 9.3 Check Work

```
[48]: df3['OFFENSE MINIMUM YEAR OR TERM'].str.contains('IND').any()
```

```
[48]: False
```

```
[49]: df3['OFFENSE MINIMUM YEAR OR TERM'].str.contains('DTH').any()
```

```
[49]: False
```

0.10.4 9.4 Changing the MIN and MAX TERM/YEAR to numeric values

The pandas method `to_numeric()` is used to change the MIN and MAX TERM/YEAR string values into float values (integers with decimal points). This will make finding future inmate sentencing

averages easier.

```
[50]: df3['OFFENSE MINIMUM YEAR OR TERM'] = pd.to_numeric(df3['OFFENSE MINIMUM YEAR_
      ↳OR TERM'])
df3['OFFENSE MINIMUM YEAR OR TERM'] = pd.to_numeric(df3['OFFENSE MINIMUM YEAR_
      ↳OR TERM'])
```

Now the value type change can be seen.

```
[51]: df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119062 entries, 0 to 129896
Data columns (total 28 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                            119062 non-null  int64
1   DATE OF BIRTH                        119062 non-null  datetime64[ns]
2   RACE DESC                            119062 non-null  object
3   GENDER                              119062 non-null  object
4   FACILITY                             27386 non-null   object
5   SENTENCE BEGIN DATE                 119062 non-null  datetime64[ns]
6   MIN TERM/YEAR                       119062 non-null  object
7   MAX TERM/YEAR                       119062 non-null  object
8   INST RELEASE DATE                   107863 non-null  datetime64[ns]
9   ACTIVE                              119062 non-null  int64
10  OFFENSE MINIMUM YEAR OR TERM         119062 non-null  float64
11  OFFENSE MAXIMUM YEAR OR TERM         119062 non-null  object
12  OFFENSE ARREST DESC                  119062 non-null  object
13  FELONY MSDMNR CODE                   105246 non-null  object
14  OFFENSE TYPE CODE                    119062 non-null  object
15  OFFENSE ATTEMPT DESC                  15577 non-null   object
16  HABITUAL CRIMINAL                    706 non-null     object
17  OFFENSE RUN CODE                      96624 non-null   object
18  COUNTY COMMITTED                     119039 non-null   object
19  OFFENSE ARREST CD                    119010 non-null   object
20  OFFENSE ARREST                       119010 non-null   object
21  OFFENSE GROUP                        119062 non-null   object
22  OFFENSE CATEGORY                      119010 non-null   object
23  SENTENCE BEGIN DATE DT                119062 non-null  datetime64[ns]
24  DATE OF BIRTH DT                     119062 non-null  datetime64[ns]
25  SENTENCE BEGIN AGE                    119062 non-null  float64
26  CURRENT AGE                          119062 non-null  float64
27  SENTENCE BEGIN YEAR                  119062 non-null  int64
dtypes: datetime64[ns](5), float64(3), int64(3), object(17)
memory usage: 26.3+ MB
```

0.11 10.0 Reducing the Database to only contain inmates incarcerated after 1979.

When looking at broad trends over the studied period, this is the range that needs to be researched. All previous incarcerations are outlier data that had incomplete digital documentation.

The DataFrame can be altered to only keep rows where inmates were incarcerated after 1979.

```
[52]: df3 = df3[df3['SENTENCE BEGIN YEAR'] > 1979]
```

The DataFrame lost around 4297 rows of inmates.

```
[53]: df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 114765 entries, 215 to 129896
Data columns (total 28 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                            114765 non-null  int64
1   DATE OF BIRTH                        114765 non-null  datetime64[ns]
2   RACE DESC                            114765 non-null  object
3   GENDER                              114765 non-null  object
4   FACILITY                             24703 non-null   object
5   SENTENCE BEGIN DATE                  114765 non-null  datetime64[ns]
6   MIN TERM/YEAR                        114765 non-null  object
7   MAX TERM/YEAR                        114765 non-null  object
8   INST RELEASE DATE                    103618 non-null  datetime64[ns]
9   ACTIVE                              114765 non-null  int64
10  OFFENSE MINIMUM YEAR OR TERM          114765 non-null  float64
11  OFFENSE MAXIMUM YEAR OR TERM          114765 non-null  object
12  OFFENSE ARREST DESC                   114765 non-null  object
13  FELONY MSDMNR CODE                    104698 non-null  object
14  OFFENSE TYPE CODE                     114765 non-null  object
15  OFFENSE ATTEMPT DESC                   15549 non-null   object
16  HABITUAL CRIMINAL                     675 non-null     object
17  OFFENSE RUN CODE                       92414 non-null   object
18  COUNTY COMMITTED                       114746 non-null  object
19  OFFENSE ARREST CD                     114735 non-null  object
20  OFFENSE ARREST                        114735 non-null  object
21  OFFENSE GROUP                          114765 non-null  object
22  OFFENSE CATEGORY                       114735 non-null  object
23  SENTENCE BEGIN DATE DT                 114765 non-null  datetime64[ns]
24  DATE OF BIRTH DT                      114765 non-null  datetime64[ns]
25  SENTENCE BEGIN AGE                     114765 non-null  float64
26  CURRENT AGE                           114765 non-null  float64
27  SENTENCE BEGIN YEAR                    114765 non-null  int64
dtypes: datetime64[ns](5), float64(3), int64(3), object(17)
memory usage: 25.4+ MB
```

0.12 11.0 Converting DataFrame to CSV for Future Use

The DataFrame method `to_csv` converts a DataFrame to a CSV file for easier storage and sharing.

```
[54]: df3.to_csv('inmate_updatedClean_offenses.csv', encoding='utf-8', index=False)
```