DataCleaning

April 12, 2021

1 Data Cleaning & Processing

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

1.1 SHR76_19 to ind_murd

[2]:
```
# Utilize Pandas dataframe to clean the data
ind_murd = pd.read_csv('RawData/SHR76_19.csv')
# What does the base data look like?
ind_murd.head()
```

[2]:
```
                  ID         CNTYFIPS      Ori    State     Agency  \
0  197603001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
1  197604001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
2  197606001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
3  197606002AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
4  197607001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage

           Agentype Source Solved  Year  StateName  …  \
0  Municipal police    FBI    Yes  1976        NaN  …
1  Municipal police    FBI    Yes  1976        NaN  …
2  Municipal police    FBI    Yes  1976        NaN  …
3  Municipal police    FBI    Yes  1976        NaN  …
4  Municipal police    FBI    Yes  1976        NaN  …

                           OffRace                 OffEthnic  \
0                            Black  Unknown or not reported
1                            White  Unknown or not reported
2                            Black  Unknown or not reported
3                            White  Unknown or not reported
4  American Indian or Alaskan Native  Unknown or not reported

                         Weapon                 Relationship  \
0  Handgun - pistol, revolver, etc  Relationship not determined
1  Handgun - pistol, revolver, etc                   Girlfriend
2  Handgun - pistol, revolver, etc                     Stranger
```

```
3  Handgun - pistol, revolver, etc      Other - known to victim
4      Knife or cutting instrument                      Brother

      Circumstance  Subcircum VicCount OffCount FileDate             MSA
0  Other arguments        NaN        0        0  30180.0  Anchorage, AK
1  Other arguments        NaN        0        0  30180.0  Anchorage, AK
2            Other        NaN        0        0  30180.0  Anchorage, AK
3  Other arguments        NaN        0        0  30180.0  Anchorage, AK
4  Other arguments        NaN        0        0  30180.0  Anchorage, AK

[5 rows x 31 columns]
```

[3]:
```
# All data attributes (columns) and types
ind_murd.dtypes
```

[3]:
```
ID                 object
CNTYFIPS           object
Ori                object
State              object
Agency             object
Agentype           object
Source             object
Solved             object
Year                int64
StateName         float64
Month              object
Incident            int64
ActionType         object
Homicide           object
Situation          object
VicAge              int64
VicSex             object
VicRace            object
VicEthnic          object
OffAge              int64
OffSex             object
OffRace            object
OffEthnic          object
Weapon             object
Relationship       object
Circumstance       object
Subcircum          object
VicCount            int64
OffCount            int64
FileDate          float64
MSA                object
dtype: object
```

```
[4]: # Data is 1976 - 2019, but only interested in data from 2010 - 2019
     recent_im = ind_murd[ind_murd['Year'] >= 2010]
     recent_im.head()
```

```
[4]:                    ID       CNTYFIPS      Ori    State     Agency  \
     626  201001001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
     627  201002001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
     628  201003001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
     629  201003002AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
     630  201003003AK00101  Anchorage, AK  AK00101  Alaska  Anchorage

                  Agentype Source Solved  Year  StateName  …  OffRace  \
     626  Municipal police    FBI     No  2010        NaN  …  Unknown
     627  Municipal police    FBI    Yes  2010        NaN  …    Black
     628  Municipal police    FBI    Yes  2010        NaN  …    Asian
     629  Municipal police    FBI    Yes  2010        NaN  …    White
     630  Municipal police    FBI    Yes  2010        NaN  …    White

                        OffEthnic                    Weapon  \
     626  Unknown or not reported  Firearm, type not stated
     627  Unknown or not reported  Firearm, type not stated
     628  Unknown or not reported  Firearm, type not stated
     629  Unknown or not reported  Firearm, type not stated
     630  Unknown or not reported  Firearm, type not stated

                      Relationship                     Circumstance  \
     626  Relationship not determined       Circumstances undetermined
     627                Acquaintance  Argument over money or property
     628                    Stranger           Felon killed by police
     629                Acquaintance  Felon killed by private citizen
     630                Acquaintance                  Other arguments

                                    Subcircum VicCount OffCount FileDate  \
     626                                  NaN        0        0  70810.0
     627                                  NaN        0        0  70810.0
     628          Felon attacked police officer        0        1  71910.0
     629  Felon killed in commission of a crime        0        0  71910.0
     630                                  NaN        0        0  71910.0

                    MSA
     626  Anchorage, AK
     627  Anchorage, AK
     628  Anchorage, AK
     629  Anchorage, AK
     630  Anchorage, AK

     [5 rows x 31 columns]
```

[5]:
```
# General use of .describe() and .unique() to see the general trends in
# given attributes
recent_im['Year'].describe()
```

[5]:
```
count    161166.000000
mean       2014.646116
std           2.851378
min        2010.000000
25%        2012.000000
50%        2015.000000
75%        2017.000000
max        2019.000000
Name: Year, dtype: float64
```

[6]:
```
recent_im['StateName'].unique()
```

[6]:
```
array([nan])
```

[7]:
```
recent_im['Source'].unique()
```

[7]:
```
array(['FBI', 'MAP'], dtype=object)
```

[8]:
```
recent_im['ActionType'].unique()
```

[8]:
```
array(['Normal update', 'Adjustment'], dtype=object)
```

[9]:
```
recent_im['Homicide'].unique()
```

[9]:
```
array(['Murder and non-negligent manslaughter',
       'Manslaughter by negligence'], dtype=object)
```

[10]:
```
recent_im['Incident'].describe()
```

[10]:
```
count    161166.000000
mean         46.481106
std         155.656687
min           0.000000
25%           1.000000
50%           2.000000
75%           8.000000
max         999.000000
Name: Incident, dtype: float64
```

[11]:
```
recent_im['Agentype'].unique()
```

[11]:
```
array(['Municipal police', 'County police', 'Primary state LE', 'Sheriff',
       'Special police', 'Tribal', 'Regional police'], dtype=object)
```

```
[12]: recent_im['OffAge'].describe()
```

```
[12]: count    161166.000000
      mean        364.614342
      std         459.063746
      min           0.000000
      25%          26.000000
      50%          41.000000
      75%         999.000000
      max         999.000000
      Name: OffAge, dtype: float64
```

```
[13]: recent_im['VicAge'].describe()
```

```
[13]: count    161166.000000
      mean         43.991617
      std          99.460189
      min           0.000000
      25%          23.000000
      50%          31.000000
      75%          44.000000
      max         999.000000
      Name: VicAge, dtype: float64
```

```
[14]: # Both Age columns use '999' to designate NaN
      recent_im['OffAge'].replace({999: np.NaN}, inplace=True)
      recent_im['OffAge'].describe()
```

```
/opt/conda/lib/python3.8/site-packages/pandas/core/series.py:4575:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().replace(
```

```
[14]: count    105798.000000
      mean         32.616902
      std          13.732891
      min           0.000000
      25%          22.000000
      50%          29.000000
      75%          40.000000
      max          99.000000
      Name: OffAge, dtype: float64
```

```
[15]: recent_im['VicAge'].replace({999: np.NaN}, inplace=True)
      recent_im['VicAge'].describe()
```

```
[15]: count    159483.000000
      mean         33.913558
      std          16.445754
      min           0.000000
      25%          22.000000
      50%          30.000000
      75%          44.000000
      max          99.000000
      Name: VicAge, dtype: float64
```

```
[16]: # Remove StateName Column because all are NaN
      # Remove Source Column because not applicable to project
      # Remove ActionType Column because not applicable to project
      # Remove Incident Column because not applicable to project and poorly defined
      # Remove FileDate Column because not useful for project
      recent_im2 = recent_im.
       ↪drop(columns=['StateName','Source','ActionType','Incident','FileDate'])
      recent_im2.head()
```

```
[16]:                   ID         CNTYFIPS      Ori   State    Agency  \
      626  201001001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
      627  201002001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
      628  201003001AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
      629  201003002AK00101  Anchorage, AK  AK00101  Alaska  Anchorage
      630  201003003AK00101  Anchorage, AK  AK00101  Alaska  Anchorage

                  Agentype Solved  Year     Month  \
      626  Municipal police     No  2010   January
      627  Municipal police    Yes  2010  February
      628  Municipal police    Yes  2010     March
      629  Municipal police    Yes  2010     March
      630  Municipal police    Yes  2010     March

                                        Homicide  …   OffSex  OffRace  \
      626  Murder and non-negligent manslaughter  …  Unknown  Unknown
      627  Murder and non-negligent manslaughter  …     Male    Black
      628  Murder and non-negligent manslaughter  …     Male    Asian
      629  Murder and non-negligent manslaughter  …   Female    White
      630  Murder and non-negligent manslaughter  …     Male    White

                        OffEthnic                   Weapon  \
      626  Unknown or not reported  Firearm, type not stated
      627  Unknown or not reported  Firearm, type not stated
      628  Unknown or not reported  Firearm, type not stated
```

```
629  Unknown or not reported  Firearm, type not stated
630  Unknown or not reported  Firearm, type not stated

                    Relationship                     Circumstance  \
626  Relationship not determined       Circumstances undetermined
627                 Acquaintance  Argument over money or property
628                     Stranger           Felon killed by police
629                 Acquaintance  Felon killed by private citizen
630                 Acquaintance                  Other arguments

                              Subcircum VicCount OffCount            MSA
626                                 NaN        0        0  Anchorage, AK
627                                 NaN        0        0  Anchorage, AK
628          Felon attacked police officer        0        1  Anchorage, AK
629  Felon killed in commission of a crime        0        0  Anchorage, AK
630                                 NaN        0        0  Anchorage, AK

[5 rows x 26 columns]
```

[17]:
```
recent_im2.dtypes
```

[17]:
```
ID                object
CNTYFIPS          object
Ori               object
State             object
Agency            object
Agentype          object
Solved            object
Year               int64
Month             object
Homicide          object
Situation         object
VicAge           float64
VicSex            object
VicRace           object
VicEthnic         object
OffAge           float64
OffSex            object
OffRace           object
OffEthnic         object
Weapon            object
Relationship      object
Circumstance      object
Subcircum         object
VicCount           int64
OffCount           int64
MSA               object
```

```
dtype: object
```

[18]:
```
# Remaining data for 2010 - 2019 is of interest and
# will be used in the project
# Save off dataframe as .csv file
recent_im2.to_csv('CleanData/ind_murd.csv', index=False)
```

1.2 UCR65_19 to total_murd

[19]:
```
# Utilize Pandas dataframe to clean the data
total_murd = pd.read_csv('RawData/UCR65_19.csv')
# What does the base data look like?
total_murd.head()
```

[19]:
```
       ORI       Name  YEAR  MRD  CLR   State         County     Agency
0  AK00101  ANCHORAGE  1965    7    6  Alaska  Anchorage, AK  Anchorage
1  AK00101  ANCHORAGE  1966   18   16  Alaska  Anchorage, AK  Anchorage
2  AK00101  ANCHORAGE  1967    1    1  Alaska  Anchorage, AK  Anchorage
3  AK00101  ANCHORAGE  1968    7    5  Alaska  Anchorage, AK  Anchorage
4  AK00101  ANCHORAGE  1969    7    4  Alaska  Anchorage, AK  Anchorage
```

[20]:
```
# Only interested in dates 2010 - 2019
total_murd['YEAR'].describe()
```

[20]:
```
count    166225.000000
mean       1992.787523
std          15.321233
min        1965.000000
25%        1980.000000
50%        1992.000000
75%        2006.000000
max        2019.000000
Name: YEAR, dtype: float64
```

[21]:
```
recent_tm = total_murd[total_murd['YEAR'] >= 2010]
recent_tm['YEAR'].describe()
```

[21]:
```
count    31629.000000
mean      2014.553922
std          2.858048
min       2010.000000
25%       2012.000000
50%       2015.000000
75%       2017.000000
max       2019.000000
Name: YEAR, dtype: float64
```

[22]:
```
# Drop Name Column because it seems redundant with Agency Column
recent_tm2 = recent_tm.drop(columns=['Name'])
recent_tm2.head()
```

[22]:
```
        ORI  YEAR  MRD  CLR   State         County     Agency
45  AK00101  2010   13   10  Alaska  Anchorage, AK  Anchorage
46  AK00101  2011   12   13  Alaska  Anchorage, AK  Anchorage
47  AK00101  2012   15   12  Alaska  Anchorage, AK  Anchorage
48  AK00101  2013   14   12  Alaska  Anchorage, AK  Anchorage
49  AK00101  2014   12    9  Alaska  Anchorage, AK  Anchorage
```

[23]:
```
# The remaining data looks useful for murder totals 2010 - 2019 by state
# Write the clean data to csv file
recent_tm2.to_csv('CleanData/total_murd.csv')
```

1.3 nst-est2019-alldata to census

[24]:
```
# Utilize Pandas dataframe to clean the data
census = pd.read_csv('RawData/nst-est2019-alldata.csv')
# What does the base data look like?
census.head(10)
```

[24]:
```
   SUMLEV REGION DIVISION  STATE              NAME  CENSUS2010POP  \
0      10      0        0      0     United States      308745538
1      20      1        0      0  Northeast Region       55317240
2      20      2        0      0    Midwest Region       66927001
3      20      3        0      0      South Region      114555744
4      20      4        0      0       West Region       71945553
5      40      3        6      1           Alabama        4779736
6      40      4        9      2            Alaska         710231
7      40      4        8      4           Arizona        6392017
8      40      3        7      5          Arkansas        2915918
9      40      4        9      6        California       37253956

   ESTIMATESBASE2010  POPESTIMATE2010  POPESTIMATE2011  POPESTIMATE2012  …  \
0          308758105        309321666        311556874        313830990  …
1           55318443         55380134         55604223         55775216  …
2           66929725         66974416         67157800         67336743  …
3          114563030        114866680        116006522        117241208  …
4           71946907         72100436         72788329         73477823  …
5            4780125          4785437          4799069          4815588  …
6             710249           713910           722128           730443  …
7            6392288          6407172          6472643          6554978  …
8            2916031          2921964          2940667          2952164  …
9           37254519         37319502         37638369         37948800  …

   RDOMESTICMIG2019  RNETMIG2011  RNETMIG2012  RNETMIG2013  RNETMIG2014  \
0          0.000000     2.493773     2.682083     2.636187     2.921500
1         -5.254530     0.887909    -0.038355    -0.469783    -0.986097
2         -2.365881    -0.963930    -0.973943    -0.006924    -0.762969
3          3.261349     5.130513     5.850458     5.292073     6.161501
4          0.614245     2.723344     3.062896     3.162262     4.026429
5          1.917501     0.578434     1.186314     1.522549     0.563489
6        -12.929847     0.587728     1.416798    -0.955359   -11.460949
7         12.609078     4.278167     6.899802     6.376679     9.168478
8          0.834503     3.294766     0.827785     0.057853    -0.091449
9         -5.151429     1.276797     1.495016     1.649031     2.203551

   RNETMIG2015  RNETMIG2016  RNETMIG2017  RNETMIG2018  RNETMIG2019
0     3.260435     3.252788     2.871957     2.153911     1.818059
1    -2.061965    -2.490484    -1.837048    -2.134447    -2.859713
2    -1.388437    -1.241784    -0.557370    -0.922755    -1.111173
3     7.277358     7.150074     6.198168     5.225519     5.203720
4     4.987285     5.261078     4.021194     3.044951     2.312083
5     0.626357     0.745172     1.090366     1.773786     2.483744
6    -7.997118    -3.897349   -10.992765   -13.859140   -12.031221
7     9.597577    11.964782    10.878879    12.962934    13.687161
8     1.075446     1.486269     2.009593     0.958896     0.923429
9     1.984957     0.500044    -0.629909    -2.130954    -3.276681

[10 rows x 151 columns]
```

[25]:
```python
# Drop sections of columns that we don't need for project
def drop_sections(name, start_year):
    cols = []
    years = 2019 - start_year + 1
    for i in range(0, years):
        attribute = name + str(start_year + i)
        cols.append(attribute)
    census.drop(cols, axis=1, inplace=True)


# Drop unnecessary columns for NPOPCHG_####
drop_sections('NPOPCHG_', 2010)

# Drop unnecessary columns for BIRTHS####
drop_sections('BIRTHS', 2010)

# Drop unnecessary columns for NATURALINC####
drop_sections('NATURALINC', 2010)

# Drop unnecessary columns for INTERNATIONALMIG####
drop_sections('INTERNATIONALMIG', 2010)
```

```
# Drop unnecessary columns for DOMESTICMIG####
drop_sections('DOMESTICMIG', 2010)

# Drop unnecessary columns for NETMIG####
drop_sections('NETMIG', 2010)

# Drop unnecessary columns for RESIDUAL####
drop_sections('RESIDUAL', 2010)

#Drop unnecessary columns for RBIRTH####
drop_sections('RBIRTH', 2011)

#Drop unnecessary columns for RDEATH####
drop_sections('RDEATH', 2011)

#Drop unnecessary columns for RNATURALINC####
drop_sections('RNATURALINC', 2011)

#Drop unnecessary columns for RINTERNATIONALMIG####
drop_sections('RINTERNATIONALMIG', 2011)

#Drop unnecessary columns for RDOMESTIC####
drop_sections('RDOMESTICMIG', 2011)

#Drop unnecessary columns for RNETMIG####
drop_sections('RNETMIG', 2011)
```

[26]:
```
# Drop columns that are not necessary for project
# Drop SUMLEV because the value is not applicable to project
# Drop DIVISION because the value is not applicable to project
# Drop STATE because its value is already in NAME column
# Drop CENSUS2010POP and ESTIMATESBASE2010
#     We will utilize the POPESTIMATES for populations sizes
census.
 ↪drop(columns=['SUMLEV','DIVISION','STATE','CENSUS2010POP','ESTIMATESBASE2010'],␣
 ↪inplace=True)
census.dtypes
```

[26]:
```
REGION               object
NAME                 object
POPESTIMATE2010       int64
POPESTIMATE2011       int64
POPESTIMATE2012       int64
POPESTIMATE2013       int64
POPESTIMATE2014       int64
POPESTIMATE2015       int64
POPESTIMATE2016       int64
```

```
POPESTIMATE2017     int64
POPESTIMATE2018     int64
POPESTIMATE2019     int64
DEATHS2010          int64
DEATHS2011          int64
DEATHS2012          int64
DEATHS2013          int64
DEATHS2014          int64
DEATHS2015          int64
DEATHS2016          int64
DEATHS2017          int64
DEATHS2018          int64
DEATHS2019          int64
dtype: object
```

[27]:
```
# Update name of REGION and NAME cols to be REGIONID and REGION
census.rename(columns={"REGION": "REGIONID", "NAME": "REGION"}, inplace=True)
census.dtypes
```

[27]:
```
REGIONID           object
REGION             object
POPESTIMATE2010     int64
POPESTIMATE2011     int64
POPESTIMATE2012     int64
POPESTIMATE2013     int64
POPESTIMATE2014     int64
POPESTIMATE2015     int64
POPESTIMATE2016     int64
POPESTIMATE2017     int64
POPESTIMATE2018     int64
POPESTIMATE2019     int64
DEATHS2010          int64
DEATHS2011          int64
DEATHS2012          int64
DEATHS2013          int64
DEATHS2014          int64
DEATHS2015          int64
DEATHS2016          int64
DEATHS2017          int64
DEATHS2018          int64
DEATHS2019          int64
dtype: object
```

[28]:
```
# This data will be sufficient for the project
# We can write it to a CleanData .csv file
census.to_csv('CleanData/census.csv', index=False)
```

1.4 crashreport

The data in this file was originally provided in a format that was difficult to manipulate as above. This data was reformatted in Google Sheets manually and was saved in its current form for use in this project.

[ ]:
```

```