Libraries

```
# libraries to read data

import numpy as np
import pandas as pd

import regex

# pip install pandas-profiling
# from https://github.com/ydataai/pandas-profiling.git

from pandas_profiling import ProfileReport

# pip install lux-api

import lux

# libraries for making graphs

import seaborn as sns

import matplotlib.pyplot as plt
import matplotlib

# libraries for maps

import os
import json
import geopandas as gpd

# libraries for data analysis

import sklearn
from sklearn.linear_model import LinearRegression
```

Set directory

```
import os
os.getcwd()
```

```
'/Users/elika_sinha/Documents/UCL/11. Dissertation/Term3'
```

```
os.chdir("/Users/elika_sinha/Documents/UCL/11. 
Dissertation/Term3/Datasets")
os.getcwd()
```

```
'/Users/elika_sinha/Documents/UCL/11. Dissertation/Term3/Datasets'
```

Datasets

City Survey

City survey cumulative data for the years-

1. 2018
2. 2019
3. 2020
4. 2021

```
city_survey_2018 = pd.read_excel('/Users/elika_sinha/Documents/UCL/11. Dissertation/Term3/Datasets/from_WCC/City Survey/2018 Westminster City Survey 2018 Full data tables 111218 FINAL.xlsx', sheet_name='Full tables 111218', header=8)
city_survey_2018.info()

city_survey_2018.sample(6, random_state=10)

city_survey_2019 = pd.read_excel('/Users/elika_sinha/Documents/UCL/11. Dissertation/Term3/Datasets/from_WCC/City Survey/2019 Westminster Council City Survey - 2019 FINAL data tables 051219.xlsx', sheet_name='Data tables', header=8)
city_survey_2019.info()

city_survey_2019.sample(6, random_state=10)

city_survey_2020 = pd.read_excel('/Users/elika_sinha/Documents/UCL/11. Dissertation/Term3/Datasets/from_WCC/City Survey/2020 Westminster City Survey 2020 - data tables 291020.xlsx', sheet_name='City Survey 2020 data tables', header=8)
city_survey_2020.info()

city_survey_2020.sample(6, random_state=10)

city_survey_2021 = pd.read_excel('/Users/elika_sinha/Documents/UCL/11. Dissertation/Term3/Datasets/from_WCC/City Survey/2021 Westminster City Survey 2021 - data tables 081121.xlsx', sheet_name='Weighted data tables', header=8)
city_survey_2021.info()

city_survey_2021.sample(6, random_state=10)
```

CACI Data

This data include income at the household level to the district level

Equalised Paycheck 2022

1. Postcode
2. OA
3. LSOA

4. MSOA
5. Districts
6. Comparators (inside London/ outside London)

Actual Paycheck 2022

1. Postcode
2. OA
3. LSOA
4. MSOA
5. Districts
6. Comparators (inside London/ outside London)

```
Equi_Paycheck22_postcode =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Equivalised Paycheck
directory 2022.xlsx', sheet_name='Postcode', header=8)
Equi_Paycheck22_postcode.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 30764 entries, 0 to 30763
Data columns (total 34 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Postcode           30764 non-null  object
 1   Large User         30764 non-null  int64
 2   Deleted Flag       30764 non-null  int64
 3   Total households   30764 non-null  int64
 4   0-5K               30764 non-null  float64
 5   5-10K              30764 non-null  float64
 6   10-15K             30764 non-null  float64
 7   15-20K             30764 non-null  float64
 8   20-25K             30764 non-null  float64
 9   25-30K             30764 non-null  float64
 10  30-35K             30764 non-null  float64
 11  35-40K             30764 non-null  float64
 12  40-45K             30764 non-null  float64
 13  45-50K             30764 non-null  float64
 14  50-55K             30764 non-null  float64
 15  55-60K             30764 non-null  float64
 16  60-65K             30764 non-null  float64
 17  65-70K             30764 non-null  float64
 18  70-75K             30764 non-null  float64
 19  75-80K             30764 non-null  float64
 20  80-85K             30764 non-null  float64
 21  85-90K             30764 non-null  float64
 22  90-95K             30764 non-null  float64
 23  95-100K            30764 non-null  float64
 24  100-120K           30764 non-null  float64
 25  120-140K          30764 non-null  float64
 26  140-160K          30764 non-null  float64
 27  160-180K          30764 non-null  float64
 28  180-200K          30764 non-null  float64
 29  200K+             30764 non-null  float64
 30  Mean Income       8153 non-null   float64
 31  Median Income     8153 non-null   float64
 32  Mode Income       8153 non-null   float64
 33  Lower Quartile    8153 non-null   float64
dtypes: float64(30), int64(3), object(1)
memory usage: 8.0+ MB
```

```
Equi_Paycheck22_postcode.sample(10, random_state=10)
```

```
       Postcode  Large User  Deleted Flag  Total households  0-5K  5-
10K  \
30696  WC2R 3DB           0             1                 0  0.00  
0.00   
13688  W 1H 5YB           0             1                 0  0.00  
0.00   
5880   SW1X 7TA           0             0                 0  0.00  
0.00   
3773   SW1P 4HU           0             0                65  0.07  
0.39   
22496  W 1W 5PR           0             0                 1  0.01  
0.03   
10328  W 1D 3RG           0             0                 0  0.00  
0.00   
24154  W 1Y 1YE           0             1                 0  0.00  
0.00   
3417   SW1P 2SH           1             1                 0  0.00  
0.00   
20629  W 1U 4AL           0             0                 4  0.00  
0.01   
6983   SW 7 1LF           1             1                 0  0.00  
0.00   

       10-15K  15-20K  20-25K  25-30K  ...  100-120K  120-140K  140-
160K  \
30696    0.00    0.00    0.00    0.00  ...      0.00      0.00  
0.00   
13688    0.00    0.00    0.00    0.00  ...      0.00      0.00  
0.00   
5880     0.00    0.00    0.00    0.00  ...      0.00      0.00  
0.00   
3773     1.49    3.01    4.61    5.77  ...      1.87      0.94  
0.42   
22496    0.08    0.12    0.14    0.14  ...      0.00      0.00  
0.00   
10328    0.00    0.00    0.00    0.00  ...      0.00      0.00  
```

```
0.00
24154    0.00    0.00    0.00    0.00  ...      0.00      0.00
0.00
3417     0.00    0.00    0.00    0.00  ...      0.00      0.00
0.00
20629    0.03    0.08    0.13    0.19  ...      0.27      0.17
0.09
6983     0.00    0.00    0.00    0.00  ...      0.00      0.00
0.00

       160-180K  180-200K  200K+  Mean Income  Median Income  Mode
Income  \
30696      0.00      0.00   0.00          NaN            NaN
NaN
13688      0.00      0.00   0.00          NaN            NaN
NaN
5880       0.00      0.00   0.00          NaN            NaN
NaN
3773       0.19      0.11   0.08      49800.0   44024.590164
37500.0
22496      0.00      0.00   0.00      32700.0   29285.714286
22500.0
10328      0.00      0.00   0.00          NaN            NaN
NaN
24154      0.00      0.00   0.00          NaN            NaN
NaN
3417       0.00      0.00   0.00          NaN            NaN
NaN
20629      0.05      0.03   0.04      67900.0   58703.703704
42500.0
6983       0.00      0.00   0.00          NaN            NaN
NaN

       Lower Quartile
30696             NaN
13688             NaN
5880              NaN
3773     30765.993266
22496    20357.142857
10328             NaN
24154             NaN
3417              NaN
20629    41206.896552
6983              NaN

[10 rows x 34 columns]
```

```python
Equi_Paycheck22_OA =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Equivalised Paycheck
directory 2022.xlsx', sheet_name='OA', header=8)
Equi_Paycheck22_OA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 783 entries, 0 to 782
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Area ID            783 non-null    object 
 1   Area Name          0 non-null      float64
 2   Total households   783 non-null    int64  
 3   0-5K               783 non-null    float64
 4   5-10K              783 non-null    float64
 5   10-15K             783 non-null    float64
 6   15-20K             783 non-null    float64
 7   20-25K             783 non-null    float64
 8   25-30K             783 non-null    float64
 9   30-35K             783 non-null    float64
 10  35-40K             783 non-null    float64
 11  40-45K             783 non-null    float64
 12  45-50K             783 non-null    float64
 13  50-55K             783 non-null    float64
 14  55-60K             783 non-null    float64
 15  60-65K             783 non-null    float64
 16  65-70K             783 non-null    float64
 17  70-75K             783 non-null    float64
 18  75-80K             783 non-null    float64
 19  80-85K             783 non-null    float64
 20  85-90K             783 non-null    float64
 21  90-95K             783 non-null    float64
 22  95-100K            783 non-null    float64
 23  100-120K           783 non-null    float64
 24  120-140K           783 non-null    float64
 25  140-160K           783 non-null    float64
 26  160-180K           783 non-null    float64
 27  180-200K           783 non-null    float64
 28  200K+              783 non-null    float64
 29  Mean Income        780 non-null    float64
 30  Median Income      780 non-null    float64
 31  Mode Income        780 non-null    float64
 32  Lower Quartile     780 non-null    float64
dtypes: float64(31), int64(1), object(1)
memory usage: 202.0+ KB
```

```
Equi_Paycheck22_OA.sample(10, random_state=10)
```

```
       Area ID  Area Name  Total households  0-5K  5-10K  10-15K  15-
20K  \
27   E00023441        NaN               211  0.78   3.43    9.89   
15.33
```

```
305  E00023738        NaN               135  0.07   0.40    1.53
3.20
579  E00024028        NaN               166  0.23   1.19    4.06
7.48
514  E00023954        NaN               158  0.22   1.14    3.90
7.12
181  E00023603        NaN               228  0.42   2.04    6.56
11.58
588  E00024038        NaN               125  0.23   1.10    3.48
5.96
195  E00023618        NaN               195  0.29   1.37    4.33
7.60
718  E00175215        NaN                63  0.20   0.88    2.57
4.08
175  E00023597        NaN               126  0.42   1.97    6.07
10.15
704  E00175201        NaN               116  0.39   1.81    5.58
9.33

     20-25K  25-30K  30-35K  ...  100-120K  120-140K  140-160K  160-
180K  \
27    18.76   19.79   18.08  ...      6.66      3.79      1.93
0.96
305    5.23    7.15    8.10  ...      8.35      5.19      2.87
1.54
579   10.76   13.02   13.23  ...      6.89      4.08      2.17
1.13
514   10.06   11.98   12.07  ...      7.18      4.37      2.39
1.27
181   16.13   19.05   19.01  ...      8.33      4.73      2.42
1.21
588    8.13    9.52    9.54  ...      5.52      3.29      1.75
0.91
195   10.75   13.13   13.68  ...      9.79      5.91      3.19
1.67
718    5.16    5.64    5.32  ...      2.10      1.21      0.63
0.32
175   13.19   14.39   13.20  ...      1.91      0.89      0.38
0.16
704   12.13   13.23   12.14  ...      1.76      0.82      0.35
0.15

     180-200K  200K+   Mean Income  Median Income  Mode Income  Lower
Quartile
27       0.63   0.63  48270.805687   40523.897059      27500.0
26152.097019
305      1.07   1.22  65155.259259   56498.297389      42500.0
39242.902208
579      0.77   0.84  55027.710843   47112.561175      37500.0
31798.941799
```

```
514      0.87   0.99  56759.240506   48132.743363      37500.0
32104.391052
181      0.79   0.81  52461.929825   44997.406639      37500.0
30320.883745
588      0.61   0.67  55955.200000   47480.490524      37500.0
31483.228512
195      1.14   1.25  59112.871795   50697.761194      42500.0
34122.807018
718      0.21   0.22  49765.079365   42290.836653      27500.0
27535.460993
175      0.09   0.07  41324.285714   36422.379827      27500.0
24886.277483
704      0.08   0.06  41266.810345   36453.687822      27500.0
24901.071723

[10 rows x 33 columns]
```

```
Equi_Paycheck22_LSOA =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Equivalised Paycheck
directory 2022.xlsx', sheet_name='LSOA', header=8)
Equi_Paycheck22_LSOA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 128 entries, 0 to 127
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Area ID            128 non-null    object
 1   Area Name          128 non-null    object
 2   Total households   128 non-null    int64
 3   0-5K               128 non-null    float64
 4   5-10K              128 non-null    float64
 5   10-15K             128 non-null    float64
 6   15-20K             128 non-null    float64
 7   20-25K             128 non-null    float64
 8   25-30K             128 non-null    float64
 9   30-35K             128 non-null    float64
 10  35-40K             128 non-null    float64
 11  40-45K             128 non-null    float64
 12  45-50K             128 non-null    float64
 13  50-55K             128 non-null    float64
 14  55-60K             128 non-null    float64
 15  60-65K             128 non-null    float64
 16  65-70K             128 non-null    float64
 17  70-75K             128 non-null    float64
 18  75-80K             128 non-null    float64
 19  80-85K             128 non-null    float64
 20  85-90K             128 non-null    float64
 21  90-95K            128 non-null    float64
 22  95-100K           128 non-null    float64
 23  100-120K          128 non-null    float64
 24  120-140K          128 non-null    float64
 25  140-160K          128 non-null    float64
 26  160-180K          128 non-null    float64
 27  180-200K          128 non-null    float64
 28  200K+             128 non-null    float64
 29  Mean Income       128 non-null    float64
 30  Median Income     128 non-null    float64
 31  Mode Income       128 non-null    int64
 32  Lower Quartile    128 non-null    float64
dtypes: float64(29), int64(2), object(2)
memory usage: 33.1+ KB
```

```
Equi_Paycheck22_LSOA.sample(10, random_state=10)
```

```
       Area ID         Area Name  Total households  0-5K  5-10K  10-
15K  \
104  E01004757  Westminster 010C               715  4.97  21.75
59.67
95   E01004748  Westminster 020E               773  0.29   1.66
6.70
41   E01004693  Westminster 017B               935  0.32   1.85
7.46
123  E01033604  Westminster 009J               751  2.89  13.40
40.87
115  E01033596  Westminster 013F               930  1.91   9.67
31.93
59   E01004712  Westminster 008C              1162  1.08   5.29
17.57
53   E01004706  Westminster 006B               797  0.37   2.10
8.08
63   E01004716  Westminster 011C               976  0.40   2.33
9.28
45   E01004697  Westminster 016D              1000  0.71   3.93
14.57
39   E01004691  Westminster 019E               986  0.41   2.38
9.41

     15-20K  20-25K  25-30K  30-35K  ...  100-120K  120-140K  140-160K
\
104   86.06   96.71   92.45   75.72  ...      5.26      2.24      0.87

95    15.00   25.99   37.18   43.64  ...     50.84     31.52     17.33

41    16.63   28.78   41.30   48.80  ...     66.71     42.88     24.47

123   67.21   85.52   91.13   81.71  ...      8.35      3.61      1.42
```

```
115   56.71   77.92   89.54   86.33  ...     26.48     14.91      7.68

59    33.10   50.41   65.77   72.33  ...     69.13     43.11     23.96

53    17.18   28.43   39.29   45.06  ...     52.89     33.27     18.57

63    20.47   35.01   49.49   57.45  ...     61.36     37.97     20.94

45    29.43   46.16   60.56   66.26  ...     55.64     33.75     18.19

39    20.69   35.26   49.72   57.63  ...     62.77     38.97     21.52


     160-180K  180-200K  200K+   Mean Income  Median Income  Mode
Income  \
104      0.34      0.17   0.10  34131.594406   29777.717685
22500
95       9.21      6.32   7.03  67046.028461   58158.553011
42500
41      13.49      9.62  11.73  70030.566845   60392.943812
110000
123      0.56      0.29   0.18  38825.419441   34557.581691
27500
115      3.93      2.64   2.96  48572.247312   41478.960396
27500
59      12.91      9.01  10.49  63989.569707   54966.838615
42500
53      10.00      6.96   7.96  67251.053952   58135.285113
42500
63      11.22      7.79   9.02  66022.110656   57069.546569
42500
45       9.50      6.40   6.81  61987.480000   53509.386383
42500
39      11.52      7.98   9.18  66265.294118   57288.367844
42500

     Lower Quartile
104    20325.716058
95     40879.244629
41     42214.383250
123    23705.565949
115    28035.514854
59     37702.946482
53     40443.776824
63     40066.496507
45     36881.214371
39     40151.433932

[10 rows x 33 columns]
```

```
Equi_Paycheck22_MSOA =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Equivalised Paycheck
directory 2022.xlsx', sheet_name='MSOA', header=8)
Equi_Paycheck22_MSOA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 33 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Area ID           24 non-null     object 
 1   Area Name         24 non-null     object 
 2   Total households  24 non-null     int64  
 3   0-5K              24 non-null     float64
 4   5-10K             24 non-null     float64
 5   10-15K            24 non-null     float64
 6   15-20K            24 non-null     float64
 7   20-25K            24 non-null     float64
 8   25-30K            24 non-null     float64
 9   30-35K            24 non-null     float64
 10  35-40K            24 non-null     float64
 11  40-45K            24 non-null     float64
 12  45-50K            24 non-null     float64
 13  50-55K            24 non-null     float64
 14  55-60K            24 non-null     float64
 15  60-65K            24 non-null     float64
 16  65-70K            24 non-null     float64
 17  70-75K            24 non-null     float64
 18  75-80K            24 non-null     float64
 19  80-85K            24 non-null     float64
 20  85-90K            24 non-null     float64
 21  90-95K            24 non-null     float64
 22  95-100K           24 non-null     float64
 23  100-120K          24 non-null     float64
 24  120-140K          24 non-null     float64
 25  140-160K          24 non-null     float64
 26  160-180K          24 non-null     float64
 27  180-200K          24 non-null     float64
 28  200K+             24 non-null     float64
 29  Mean Income       24 non-null     float64
 30  Median Income     24 non-null     float64
 31  Mode Income       24 non-null     int64  
 32  Lower Quartile    24 non-null     float64
dtypes: float64(29), int64(2), object(2)
memory usage: 6.3+ KB
```

```
Equi_Paycheck22_MSOA.sample(10, random_state=10)
```

```
      Area ID        Area Name  Total households   0-5K   5-10K  10-
15K  \
5   E02000965  Westminster 006              4317   5.50   27.94
94.68
1   E02000961  Westminster 002              5085   6.62   33.67
113.38
7   E02000967  Westminster 008              5414   3.38   18.25
67.26
2   E02000962  Westminster 003              5621  13.13   55.48
158.70
11  E02000971  Westminster 012              5324   5.03   26.57
94.17
12  E02000972  Westminster 013              6109   9.67   50.13
169.84
21  E02000981  Westminster 022              5632   5.56   28.62
98.73
13  E02000973  Westminster 014              6150   5.15   27.99
101.68
14  E02000974  Westminster 015              5571   5.70   29.50
102.14
3   E02000963  Westminster 004              4411  24.45  105.47
298.02

    15-20K  20-25K  25-30K  30-35K  ...  100-120K  120-140K  140-160K
\
5   176.55  258.92  320.47  333.21  ...    185.96    107.49     55.65

1   209.13  303.25  372.10  385.05  ...    226.51    132.06     68.95

7   138.53  224.27  303.95  341.89  ...    314.70    190.28    102.19

2   257.22  342.87  402.25  408.62  ...    255.64    150.46     79.06

11  183.99  281.44  361.06  386.56  ...    256.38    151.15     79.74

12  309.31  436.24  516.24  513.68  ...    227.60    134.03     71.61

21  188.38  283.91  362.03  387.84  ...    295.09    178.52     96.43

13  201.72  311.08  401.44  432.29  ...    313.03    187.00     99.81

14  195.85  296.38  378.41  404.57  ...    265.08    155.04     81.10

3   455.51  545.81  553.74  477.15  ...     39.34     16.68      6.43


    160-180K  180-200K  200K+   Mean Income  Median Income  Mode
Income  \
5      28.18     18.53  19.07  55965.795691   48318.968152
```

```
37500
1      35.22     23.39  24.88  56553.238938   48727.148201
42500
7      53.21     35.81  38.22  63301.219062   54856.932677
42500
2      40.49     26.88  28.10  56354.584593   48561.603167
42500
11     41.07     27.46  29.40  58645.627348   50633.183199
42500
12     37.66     25.89  29.93  53116.976592   44988.054309
37500
21     50.66     34.50  38.08  60482.546165   52063.456062
42500
13     51.89     34.99  37.89  59910.551220   51685.052086
42500
14     41.44     27.49  28.92  58312.387363   50466.823106
42500
3       2.51      1.29   0.83  36503.006121   32331.551923
27500

    Lower Quartile
5     32928.933705
1     33026.879626
7     38214.169115
2     32148.695610
11    34898.851407
12    30348.660645
21    35609.441566
13    35579.272067
14    34755.295746
3     22008.940840

[10 rows x 33 columns]
```

```
Equi_Paycheck22_Ward =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Equivalised Paycheck
directory 2022.xlsx', sheet_name='Ward', header=8)
Equi_Paycheck22_Ward.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 33 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Area ID           20 non-null     object
 1   Area Name         20 non-null     object
 2   Total households  20 non-null     int64
 3   0-5K              20 non-null     float64
 4   5-10K            20 non-null     float64
 5   10-15K           20 non-null     float64
 6   15-20K           20 non-null     float64
 7   20-25K           20 non-null     float64
 8   25-30K           20 non-null     float64
 9   30-35K           20 non-null     float64
 10  35-40K           20 non-null     float64
 11  40-45K           20 non-null     float64
 12  45-50K           20 non-null     float64
 13  50-55K           20 non-null     float64
 14  55-60K           20 non-null     float64
 15  60-65K           20 non-null     float64
 16  65-70K           20 non-null     float64
 17  70-75K           20 non-null     float64
 18  75-80K           20 non-null     float64
 19  80-85K           20 non-null     float64
 20  85-90K           20 non-null     float64
 21  90-95K           20 non-null     float64
 22  95-100K          20 non-null     float64
 23  100-120K         20 non-null     float64
 24  120-140K         20 non-null     float64
 25  140-160K         20 non-null     float64
 26  160-180K         20 non-null     float64
 27  180-200K         20 non-null     float64
 28  200K+            20 non-null     float64
 29  Mean Income      20 non-null     float64
 30  Median Income    20 non-null     float64
 31  Mode Income      20 non-null     int64
 32  Lower Quartile   20 non-null     float64
dtypes: float64(29), int64(2), object(2)
memory usage: 5.3+ KB
```

```
Equi_Paycheck22_Ward.sample(10, random_state=10)
```

```
      Area ID                    Area Name  Total households   0-5K
5-10K  \
7   E05000637  Knightsbridge and Belgravia              4906   2.22
12.55
10  E05000640                   Maida Vale              4930   8.25
41.34
5   E05000635                  Harrow Road              5881  18.13
86.77
6   E05000636                    Hyde Park              7995   7.31
37.93
3   E05000633                    Churchill              5291  17.83
83.27
18  E05000648                   Westbourne              6139  32.01
140.97
13  E05000643                Regent's Park              7955  14.76
64.37
2   E05000632  Bryanston and Dorset Square              7764   6.53
```

```
34.89
14  E05000644                   St James's              6223   7.07
36.97
8   E05000638               Lancaster Gate              8098   6.08
32.93

    10-15K  15-20K  20-25K  25-30K  30-35K  ...  100-120K  120-140K  
140-160K  \
7    48.72  105.45  178.09  249.87  288.89  ...    308.15    190.29   
104.62
10  135.92  242.54  338.68  399.83  399.07  ...    196.37    114.49   
59.91
5   270.12  453.31  592.06  649.95  601.25  ...    102.01     49.34   
21.65
6   133.28  260.40  400.72  518.92  561.39  ...    402.88    238.30   
125.81
3   250.62  404.52  510.12  545.35  496.34  ...    138.65     81.19   
43.48
18  398.91  602.44  711.33  714.69  615.13  ...     97.89     50.21   
23.38
13  191.46  323.85  448.81  543.22  564.66  ...    380.85    223.87   
117.16
2   125.54  249.43  387.59  504.29  546.54  ...    391.87    231.96   
122.55
14  127.51  238.45  347.72  427.77  443.74  ...    318.30    198.01   
110.40
8   119.44  238.09  371.42  486.76  532.82  ...    452.48    276.24   
150.38

    160-180K  180-200K  200K+   Mean Income  Median Income  Mode  
Income  \
7      55.82     38.62  44.44  65874.971464   56992.089671        
42500
10     30.66     20.38  21.49  53922.926978   46016.250000        
37500
5       9.50      5.45   4.30  42437.871110   37305.390762        
27500
6      64.71     43.13  45.46  59641.707317   51629.461963        
42500
3      23.00     15.92  18.67  46150.544321   38527.819014        
27500
18     10.81      6.52   5.66  39636.541782   33813.421553        
27500
13     59.65     39.28  40.09  57705.239472   49935.145257        
42500
2      63.08     42.09  44.57  59724.327666   51673.474120        
42500
14     59.83     42.06  49.78  59877.436928   50799.847467        
42500
8      79.51     54.47  60.92  62289.063966   53656.357865        
```

```
42500

    Lower Quartile
7     39950.741064
10    30826.170847
5     25383.567967
6     35623.486779
3     25517.007426
18    22533.423306
13    33562.143591
2     35700.253485
14    34172.037680
8     36951.187379

[10 rows x 33 columns]
```

```
Equi_Paycheck22_District =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Equivalised Paycheck
directory 2022.xlsx', sheet_name='District', header=8)
Equi_Paycheck22_District.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 33 entries, 0 to 32
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Area ID            33 non-null     object
 1   Area Name          33 non-null     object
 2   Total households   33 non-null     int64
 3   0-5K               33 non-null     float64
 4   5-10K              33 non-null     float64
 5   10-15K             33 non-null     float64
 6   15-20K             33 non-null     float64
 7   20-25K             33 non-null     float64
 8   25-30K             33 non-null     float64
 9   30-35K             33 non-null     float64
 10  35-40K             33 non-null     float64
 11  40-45K             33 non-null     float64
 12  45-50K             33 non-null     float64
 13  50-55K             33 non-null     float64
 14  55-60K             33 non-null     float64
 15  60-65K             33 non-null     float64
 16  65-70K             33 non-null     float64
 17  70-75K             33 non-null     float64
 18  75-80K             33 non-null     float64
 19  80-85K             33 non-null     float64
 20  85-90K             33 non-null     float64
 21  90-95K             33 non-null     float64
 22  95-100K           33 non-null     float64
 23  100-120K          33 non-null     float64
 24  120-140K          33 non-null     float64
 25  140-160K          33 non-null     float64
 26  160-180K          33 non-null     float64
 27  180-200K          33 non-null     float64
 28  200K+             33 non-null     float64
 29  Mean Income       33 non-null     float64
 30  Median Income     33 non-null     float64
 31  Mode Income       33 non-null     int64
 32  Lower Quartile    33 non-null     float64
dtypes: float64(29), int64(2), object(2)
memory usage: 8.6+ KB
```

```
Equi_Paycheck22_District.sample(10, random_state=10)
```

```
      Area ID               Area Name  Total households     0-5K    5-
10K  \
21  E09000022                 Lambeth            136626   474.44
2139.47
7   E09000008                 Croydon            156613   954.33
3874.83
5   E09000006                 Bromley            141733   478.14
2046.04
2   E09000003                  Barnet            153994   429.85
1966.14
3   E09000004                  Bexley            100651   618.54
2311.18
22  E09000023                Lewisham            131651   788.10
3301.43
13  E09000014                Haringey            106832   606.84
2521.31
24  E09000025                  Newham            118000  1371.05
5215.00
12  E09000013  Hammersmith and Fulham             79961   208.44
972.39
1   E09000002    Barking and Dagenham             79124   637.55
2646.43

      10-15K    15-20K    20-25K    25-30K    30-35K  ...  100-120K
120-140K  \
21   6316.11  10150.41  12908.83  14027.96  13009.44  ...   3100.57
1624.39
7   10556.58  15717.36  18546.82  18761.68  16275.60  ...   1758.26
788.87
5    6077.68  10056.91  13165.43  14660.27  13850.36  ...   2853.76
1378.29
2    6028.70  10176.41  13575.10  15379.27  14746.96  ...   3444.70
1698.74
3    6267.98   9650.02  11775.93  12228.14  10798.82  ...    986.99
```

```
412.75
22   9040.03  13354.12  15577.50  15589.30  13420.71  ...   1669.16
795.03
13   6851.68  10094.94  11809.01  11906.64  10369.99  ...   1884.79
975.30
24  12596.65  16282.12  16774.87  14964.95  11611.70  ...    820.72
393.94
12   2960.64   4903.73   6432.49   7226.28   6938.87  ...   2536.37
1404.16
1    7176.41  10356.87  11566.16  10888.44   8719.35  ...    296.63
108.70

    140-160K  160-180K  180-200K   200K+   Mean Income  Median Income
\
21    771.89    364.41    225.04  208.07  44733.260141   38639.349942

7     320.37    130.42     69.21   46.85  37636.861308   33039.795768

5     597.12    256.55    143.08  105.63  44066.173368   38829.135647

2     752.40    330.53    188.94  147.91  45436.474603   39949.326441

3     155.64     59.06     29.21   17.44  37546.246932   33460.429010

22    345.57    150.89     86.47   69.24  38100.157993   33045.673441

13    458.05    213.81    130.44  117.37  40645.940636   34641.074871

24    174.62     77.89     45.54   37.23  31842.621356   27258.714530

12    698.49    341.54    217.18  209.53  49459.830042   42461.927409

1      36.60     12.70      5.81    3.11  31640.392169   28296.422628


    Mode Income  Lower Quartile
21        27500    25772.471550
7         27500    22170.223790
5         27500    26230.894793
2         27500    27055.461670
3         27500    22681.329627
22        27500    22063.575670
13        27500    22808.546186
24        22500    18168.291353
12        27500    28122.325733
1         22500    19499.723372

[10 rows x 33 columns]
```

```
Equi_Paycheck22_Comparators =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Equivalised Paycheck
directory 2022.xlsx', sheet_name='Comparators', header=8)
Equi_Paycheck22_Comparators.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 33 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Area ID           5 non-null      object
 1   Area Name         5 non-null      object
 2   Total households  5 non-null      int64
 3   0-5K              5 non-null      float64
 4   5-10K             5 non-null      float64
 5   10-15K            5 non-null      float64
 6   15-20K            5 non-null      float64
 7   20-25K            5 non-null      float64
 8   25-30K            5 non-null      float64
 9   30-35K            5 non-null      float64
 10  35-40K            5 non-null      float64
 11  40-45K            5 non-null      float64
 12  45-50K            5 non-null      float64
 13  50-55K            5 non-null      float64
 14  55-60K            5 non-null      float64
 15  60-65K            5 non-null      float64
 16  65-70K            5 non-null      float64
 17  70-75K            5 non-null      float64
 18  75-80K            5 non-null      float64
 19  80-85K            5 non-null      float64
 20  85-90K            5 non-null      float64
 21  90-95K            5 non-null      float64
 22  95-100K           5 non-null      float64
 23  100-120K          5 non-null      float64
 24  120-140K          5 non-null      float64
 25  140-160K          5 non-null      float64
 26  160-180K          5 non-null      float64
 27  180-200K          5 non-null      float64
 28  200K+             5 non-null      float64
 29  Mean Income       5 non-null      float64
 30  Median Income     5 non-null      float64
 31  Mode Income       5 non-null      int64
 32  Lower Quartile    5 non-null      float64
dtypes: float64(29), int64(2), object(2)
memory usage: 1.4+ KB
```

```
Equi_Paycheck22_Comparators.sample(5, random_state=10)
```

```
          Area ID       Area Name  Total households       0-5K      5-
10K  \
2       E12000007          London           3597485   17520.46
73094.31
3   Great Britain   Great Britain          27745087  238109.89
807665.14
0       E13000001    Inner London           1544423    6923.74
29186.10
4  United Kingdom  United Kingdom          28491961  247469.50
839389.87
1       E13000002    Outer London           2053062   10596.72
43908.21

       10-15K      15-20K      20-25K      25-30K      30-35K  ...
100-120K  \
2   202241.09   306873.59   371316.40   386844.56   346608.10  ...
71508.29
3  2008162.35  2859494.90  3277306.01  3253342.29  2792868.44  ...
341925.03
0    80426.21   121327.53   146804.53   153954.66   139634.27  ...
40024.75
4  2084379.48  2961370.25  3384957.05  3351029.76  2869552.39  ...
345584.52
1   121814.89   185546.06   224511.87   232889.89   206973.82  ...
31483.54

    120-140K  140-160K  160-180K  180-200K     200K+   Mean Income  \
2   37564.04  17980.79   8574.06   5362.47   5130.54  42494.488466
3  162956.30  70943.73  31058.29  17874.74  14588.96  37572.544652
0   22299.53  11250.26   5602.06   3641.56   3718.14  45506.826705
4  164483.79  71534.55  31292.55  17997.82  14674.50  37405.713335
1   15264.51   6730.54   2971.99   1720.91   1412.39  40228.446593

   Median Income  Mode Income  Lower Quartile
2   36426.814264        27500    24034.858143
3   32557.340152        22500    21560.488198
0   38470.741322        27500    25046.690370
4   32399.999029        22500    21462.915386
1   35007.427605        27500    23371.750901

[5 rows x 33 columns]
```

```
Paycheck22_postcode =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Paycheck directory
2022.xlsx', sheet_name='Postcode', header=8)
Paycheck22_postcode.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 30764 entries, 0 to 30763
Data columns (total 34 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Postcode          30764 non-null  object 
 1   Large User        30764 non-null  int64  
 2   Deleted Flag      30764 non-null  int64  
 3   Total households  30764 non-null  int64  
 4   0-5K              30764 non-null  float64
 5   5-10K             30764 non-null  float64
 6   10-15K            30764 non-null  float64
 7   15-20K            30764 non-null  float64
 8   20-25K            30764 non-null  float64
 9   25-30K            30764 non-null  float64
 10  30-35K            30764 non-null  float64
 11  35-40K            30764 non-null  float64
 12  40-45K            30764 non-null  float64
 13  45-50K            30764 non-null  float64
 14  50-55K            30764 non-null  float64
 15  55-60K            30764 non-null  float64
 16  60-65K            30764 non-null  float64
 17  65-70K            30764 non-null  float64
 18  70-75K            30764 non-null  float64
 19  75-80K            30764 non-null  float64
 20  80-85K            30764 non-null  float64
 21  85-90K            30764 non-null  float64
 22  90-95K            30764 non-null  float64
 23  95-100K           30764 non-null  float64
 24  100-120K          30764 non-null  float64
 25  120-140K          30764 non-null  float64
 26  140-160K          30764 non-null  float64
 27  160-180K          30764 non-null  float64
 28  180-200K          30764 non-null  float64
 29  200K+             30764 non-null  float64
 30  Mean Income       8153 non-null   float64
 31  Median Income     8153 non-null   float64
 32  Mode Income       8153 non-null   float64
 33  Lower Quartile    8153 non-null   float64
dtypes: float64(30), int64(3), object(1)
memory usage: 8.0+ MB
```

```
Paycheck22_postcode.sample(5, random_state=10)
```

```
      Postcode  Large User  Deleted Flag  Total households  0-5K  5-
10K  \
30696  WC2R 3DB           0             1                 0   0.0   
0.0   
13688  W 1D 2DR           0             0                 0   0.0   
0.0   
5880   SW1W 9RA           1             1                 0   0.0   
```

```
0.0
3773   SW1P 3AS           1             1                 0   0.0
0.0
22496  W 1S 1DY           0             1                 0   0.0
0.0

       10-15K  15-20K  20-25K  25-30K  ...  100-120K  120-140K  140-
160K  \
30696     0.0     0.0     0.0     0.0  ...       0.0       0.0
0.0
13688     0.0     0.0     0.0     0.0  ...       0.0       0.0
0.0
5880      0.0     0.0     0.0     0.0  ...       0.0       0.0
0.0
3773      0.0     0.0     0.0     0.0  ...       0.0       0.0
0.0
22496     0.0     0.0     0.0     0.0  ...       0.0       0.0
0.0

       160-180K  180-200K  200K+  Mean Income  Median Income  Mode
Income  \
30696       0.0       0.0    0.0          NaN            NaN
NaN
13688       0.0       0.0    0.0          NaN            NaN
NaN
5880        0.0       0.0    0.0          NaN            NaN
NaN
3773        0.0       0.0    0.0          NaN            NaN
NaN
22496       0.0       0.0    0.0          NaN            NaN
NaN

       Lower Quartile
30696             NaN
13688             NaN
5880              NaN
3773              NaN
22496             NaN

[5 rows x 34 columns]
```

```
Paycheck22_OA = pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Paycheck directory
2022.xlsx', sheet_name='OA', header=8)
Paycheck22_OA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 783 entries, 0 to 782
Data columns (total 33 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Area ID           783 non-null    object 
 1   Area Name         0 non-null      float64
 2   Total households  783 non-null    int64  
 3   0-5K              783 non-null    float64
 4   5-10K             783 non-null    float64
 5   10-15K            783 non-null    float64
 6   15-20K            783 non-null    float64
 7   20-25K            783 non-null    float64
 8   25-30K            783 non-null    float64
 9   30-35K            783 non-null    float64
 10  35-40K            783 non-null    float64
 11  40-45K            783 non-null    float64
 12  45-50K            783 non-null    float64
 13  50-55K            783 non-null    float64
 14  55-60K            783 non-null    float64
 15  60-65K            783 non-null    float64
 16  65-70K            783 non-null    float64
 17  70-75K            783 non-null    float64
 18  75-80K            783 non-null    float64
 19  80-85K            783 non-null    float64
 20  85-90K            783 non-null    float64
 21  90-95K            783 non-null    float64
 22  95-100K           783 non-null    float64
 23  100-120K          783 non-null    float64
 24  120-140K          783 non-null    float64
 25  140-160K          783 non-null    float64
 26  160-180K          783 non-null    float64
 27  180-200K          783 non-null    float64
 28  200K+             783 non-null    float64
 29  Mean Income       779 non-null    float64
 30  Median Income     779 non-null    float64
 31  Mode Income       779 non-null    float64
 32  Lower Quartile    779 non-null    float64
dtypes: float64(31), int64(1), object(1)
memory usage: 202.0+ KB
```

```
Paycheck22_OA.sample(5, random_state=10)
```

```
       Area ID  Area Name  Total households  0-5K  5-10K  10-15K  15-
20K  \
27   E00023437        NaN               110  0.09   0.66    1.73   
2.95   
305  E00023732        NaN               131  0.13   0.92    2.37   
3.98   
579  E00024023        NaN               184  0.61   3.64    7.60  
10.72   
514  E00023949        NaN               136  0.70   4.00    7.99  
10.77   
181  E00023599        NaN               127  0.76   4.10    7.81  
```

```
10.21

     20-25K  25-30K  30-35K  ...  100-120K  120-140K  140-160K  160-
180K  \
27     3.77    4.57    4.80  ...     11.17      6.91      3.28  
2.29  
305    5.03    6.04    6.28  ...     12.18      7.11      3.18  
2.09  
579   11.79   12.65   12.00  ...     10.62      5.41      2.16  
1.29  
514   11.30   11.56   10.46  ...      4.85      2.13      0.74  
0.38  
181   10.51   10.65    9.60  ...      4.47      1.91      0.64  
0.32  

     180-200K  200K+   Mean Income  Median Income  Mode Income  Lower 
Quartile  
27       1.32   0.99  74502.727273   67158.203125     110000.0  
43004.966887  
305      1.11   0.69  70203.816794   63118.361153     110000.0  
40721.649485  
579      0.62   0.34  56000.543478   48666.069830      42500.0  
29600.790514  
514      0.16   0.06  47063.235294   40568.295115      27500.0  
24663.716814  
181      0.12   0.05  46676.377953   40325.027086      27500.0  
24219.790676  

[5 rows x 33 columns]
```

```
Paycheck22_LSOA = pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Paycheck directory
2022.xlsx', sheet_name='LSOA', header=8)
Paycheck22_LSOA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 128 entries, 0 to 127
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Area ID            128 non-null    object 
 1   Area Name          128 non-null    object 
 2   Total households   128 non-null    int64  
 3   0-5K               128 non-null    float64
 4   5-10K              128 non-null    float64
 5   10-15K             128 non-null    float64
 6   15-20K             128 non-null    float64
 7   20-25K             128 non-null    float64
 8   25-30K             128 non-null    float64
 9   30-35K            128 non-null    float64
 10  35-40K            128 non-null    float64
 11  40-45K            128 non-null    float64
 12  45-50K            128 non-null    float64
 13  50-55K            128 non-null    float64
 14  55-60K            128 non-null    float64
 15  60-65K            128 non-null    float64
 16  65-70K            128 non-null    float64
 17  70-75K            128 non-null    float64
 18  75-80K            128 non-null    float64
 19  80-85K            128 non-null    float64
 20  85-90K            128 non-null    float64
 21  90-95K            128 non-null    float64
 22  95-100K           128 non-null    float64
 23  100-120K          128 non-null    float64
 24  120-140K          128 non-null    float64
 25  140-160K          128 non-null    float64
 26  160-180K          128 non-null    float64
 27  180-200K          128 non-null    float64
 28  200K+             128 non-null    float64
 29  Mean Income       128 non-null    float64
 30  Median Income     128 non-null    float64
 31  Mode Income       128 non-null    int64  
 32  Lower Quartile    128 non-null    float64
dtypes: float64(29), int64(2), object(2)
memory usage: 33.1+ KB
```

```
Paycheck22_LSOA.sample(5, random_state=10)
```

```
       Area ID         Area Name  Total households  0-5K  5-10K  10-
15K  \
104  E01004757  Westminster 010C               715  5.93  29.62   
52.83   
95   E01004748  Westminster 020E               773  0.83   5.73   
13.99   
41   E01004693  Westminster 017B               935  0.73   5.18   
13.08   
123  E01033604  Westminster 009J               751  6.84  34.70   
61.93   
115  E01033596  Westminster 013F               930  5.61  30.63   
58.39   

     15-20K  20-25K  25-30K  30-35K  ...  100-120K  120-140K  140-160K  
\
104   65.46   64.53   63.02   55.02  ...     21.70      9.40      3.23   

95    22.65   27.91   33.02   34.09  ...     78.02     48.67     23.22   

41    21.89   27.80   33.80   35.76  ...    105.36     68.33     33.61   
```

```
123   76.10   74.08   71.29   61.24  ...     16.03      6.09      1.84

115   75.82   77.34   77.58   69.22  ...     35.70     17.41      6.90


     160-180K  180-200K  200K+   Mean Income  Median Income  Mode
Income  \
104      1.67      0.68   0.29  43789.636364   37043.604651
17500
95      16.24      9.26   6.61  73954.695990   66604.202158
110000
41      24.11     14.11  10.56  78975.700535   72349.818747
110000
123      0.85      0.30   0.11  40155.113182   34128.020901
17500
115      4.22      2.15   1.42  47863.752688   40326.302919
27500

     Lower Quartile
104    21930.110026
95     42116.527943
41     46045.525552
123    20552.105832
115    24011.507629

[5 rows x 33 columns]
```

```
Paycheck22_MSOA = pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Paycheck directory
2022.xlsx', sheet_name='MSOA', header=8)
Paycheck22_MSOA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 33 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Area ID           24 non-null     object
 1   Area Name         24 non-null     object
 2   Total households  24 non-null     int64
 3   0-5K              24 non-null     float64
 4   5-10K             24 non-null     float64
 5   10-15K            24 non-null     float64
 6   15-20K            24 non-null     float64
 7   20-25K            24 non-null     float64
 8   25-30K            24 non-null     float64
 9   30-35K            24 non-null     float64
 10  35-40K            24 non-null     float64
 11  40-45K            24 non-null     float64
 12  45-50K            24 non-null     float64
 13  50-55K            24 non-null     float64
 14  55-60K            24 non-null     float64
 15  60-65K            24 non-null     float64
 16  65-70K            24 non-null     float64
 17  70-75K            24 non-null     float64
 18  75-80K            24 non-null     float64
 19  80-85K            24 non-null     float64
 20  85-90K            24 non-null     float64
 21  90-95K            24 non-null     float64
 22  95-100K           24 non-null     float64
 23  100-120K          24 non-null     float64
 24  120-140K          24 non-null     float64
 25  140-160K          24 non-null     float64
 26  160-180K          24 non-null     float64
 27  180-200K          24 non-null     float64
 28  200K+             24 non-null     float64
 29  Mean Income       24 non-null     float64
 30  Median Income     24 non-null     float64
 31  Mode Income       24 non-null     int64
 32  Lower Quartile    24 non-null     float64
dtypes: float64(29), int64(2), object(2)
memory usage: 6.3+ KB
```

```
Paycheck22_MSOA.sample(5, random_state=10)
```

```
      Area ID        Area Name  Total households   0-5K   5-10K  10-
15K  \
5   E02000965  Westminster 006             4317  11.44   68.40
144.86
1   E02000961  Westminster 002             5085  14.33   81.40
164.13
7   E02000967  Westminster 008             5414   9.22   57.37
128.48
2   E02000962  Westminster 003             5621  40.05  159.90
251.39
11  E02000971  Westminster 012             5324  13.78   82.45
175.08

    15-20K  20-25K  25-30K  30-35K  ...  100-120K  120-140K  140-160K
\
5   208.53  233.93  256.05  247.55  ...    319.23    181.43     80.11

1   227.80  249.64  269.68  259.29  ...    437.11    274.48    133.02

7   195.65  230.80  264.05  265.37  ...    490.86    296.52    137.89

2   304.87  312.26  325.40  306.14  ...    411.72    242.00    110.47

11  252.75  284.24  311.80  302.01  ...    400.35    230.10    102.83
```

```
    160-180K  180-200K  200K+   Mean Income  Median Income  Mode
Income  \
5      52.44     27.79  17.69  62495.719249   54204.175333
110000
1      95.08     55.97  43.70  68024.190757   59056.201297
110000
7      94.52     52.81  37.17  69603.191725   61813.542154
110000
2      74.79     41.39  29.02  61455.059598   52851.529381
110000
11     68.20     36.79  24.39  63146.074380   54687.070481
110000

    Lower Quartile
5     33151.686528
1     35093.167702
7     38576.018643
2     30185.862677
11    33491.606238

[5 rows x 33 columns]
```

```
Paycheck22_Districts =
pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Paycheck directory
2022.xlsx', sheet_name='District', header=8)
Paycheck22_Districts.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 33 entries, 0 to 32
Data columns (total 33 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Area ID           33 non-null     object
 1   Area Name         33 non-null     object
 2   Total households  33 non-null     int64
 3   0-5K              33 non-null     float64
 4   5-10K             33 non-null     float64
 5   10-15K            33 non-null     float64
 6   15-20K            33 non-null     float64
 7   20-25K            33 non-null     float64
 8   25-30K            33 non-null     float64
 9   30-35K            33 non-null     float64
 10  35-40K            33 non-null     float64
 11  40-45K            33 non-null     float64
 12  45-50K            33 non-null     float64
 13  50-55K            33 non-null     float64
 14  55-60K            33 non-null     float64
 15  60-65K            33 non-null     float64
 16  65-70K            33 non-null     float64
 17  70-75K            33 non-null     float64
 18  75-80K            33 non-null     float64
 19  80-85K            33 non-null     float64
 20  85-90K            33 non-null     float64
 21  90-95K            33 non-null     float64
 22  95-100K           33 non-null     float64
 23  100-120K          33 non-null     float64
 24  120-140K          33 non-null     float64
 25  140-160K          33 non-null     float64
 26  160-180K          33 non-null     float64
 27  180-200K          33 non-null     float64
 28  200K+             33 non-null     float64
 29  Mean Income       33 non-null     float64
 30  Median Income     33 non-null     float64
 31  Mode Income       33 non-null     int64
 32  Lower Quartile    33 non-null     float64
dtypes: float64(29), int64(2), object(2)
memory usage: 8.6+ KB
```

```
Paycheck22_Districts.sample(5, random_state=10)
```

```
      Area ID Area Name  Total households     0-5K    5-10K    10-15K  
\
21  E09000022   Lambeth            136626   887.68  4475.10   8124.34   

7   E09000008   Croydon            156613  1554.01  6883.36  11339.80   

5   E09000006   Bromley            141733   920.76  4344.25   7598.07   

2   E09000003    Barnet            153994   690.28  3534.46   6662.85   

3   E09000004    Bexley            100651   993.94  4225.47   6827.54   


      15-20K    20-25K    25-30K    30-35K  ...  100-120K  120-140K  
140-160K  \
21  10300.65  10421.99  10466.57   9405.33  ...   6491.04   3368.60   
1386.92   
7   13476.00  13042.96  12673.42  11094.57  ...   6141.68   3041.57   
1198.58   
5    9529.23   9683.71   9840.85   8978.43  ...   8170.13   4478.94   
1921.66   
2    8866.26   9423.71   9916.09   9301.23  ...  10149.45   5703.22   
2500.63   
3    8103.08   7906.53   7775.91   6898.20  ...   4380.02   2168.12   
840.27   
```

```
    160-180K  180-200K   200K+   Mean Income  Median Income  Mode 
Income  \
21    860.04    432.45  258.27  51011.091154   42819.116174        
27500   
7     713.53    342.05  188.66  46834.313882   38930.584181        
17500   
5    1226.92    631.59  381.54  54920.722062   46349.012110        
27500   
2    1629.28    859.73  545.04  58846.317973   50466.196463       
110000   
3     485.34    220.98  108.82  48490.728855   40748.490778        
17500   

    Lower Quartile  
21    24974.448258  
7     22261.787202  
5     26705.762206  
2     29699.906919  
3     23169.987340  

[5 rows x 33 columns]
```

```
Paycheck22_Comparators = 
pd.read_excel('/Users/elika_sinha/Documents/UCL/11. 
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data 
(Income)/Westminster City Council - Westminster - Paycheck directory 
2022.xlsx', sheet_name='Comparators', header=8)
Paycheck22_Comparators.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 33 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Area ID           5 non-null      object 
 1   Area Name         5 non-null      object 
 2   Total households  5 non-null      int64  
 3   0-5K              5 non-null      float64
 4   5-10K             5 non-null      float64
 5   10-15K            5 non-null      float64
 6   15-20K            5 non-null      float64
 7   20-25K            5 non-null      float64
 8   25-30K            5 non-null      float64
 9   30-35K            5 non-null      float64
 10  35-40K            5 non-null      float64
 11  40-45K            5 non-null      float64
 12  45-50K            5 non-null      float64
 13  50-55K            5 non-null      float64
 14  55-60K            5 non-null      float64
 15  60-65K            5 non-null      float64
 16  65-70K            5 non-null      float64
 17  70-75K            5 non-null      float64
 18  75-80K            5 non-null      float64
 19  80-85K            5 non-null      float64
 20  85-90K            5 non-null      float64
 21  90-95K            5 non-null      float64
 22  95-100K           5 non-null      float64
 23  100-120K          5 non-null      float64
 24  120-140K          5 non-null      float64
 25  140-160K          5 non-null      float64
 26  160-180K          5 non-null      float64
 27  180-200K          5 non-null      float64
 28  200K+             5 non-null      float64
 29  Mean Income       5 non-null      float64
 30  Median Income     5 non-null      float64
 31  Mode Income       5 non-null      int64
 32  Lower Quartile    5 non-null      float64
dtypes: float64(29), int64(2), object(2)
memory usage: 1.4+ KB
```

```
Paycheck22_Comparators.sample(5, random_state=10)
```

```
          Area ID       Area Name  Total households       0-5K
5-10K  \
2       E12000007          London           3597485   26070.98
122749.70
3   Great Britain   Great Britain          27745087  399726.88
1541319.92
0       E13000001    Inner London           1544423   10793.44
52193.73
4  United Kingdom  United Kingdom          28491961  416945.19
1604093.89
1       E13000002    Outer London           2053062   15277.54
70555.97

       10-15K      15-20K      20-25K      25-30K      30-35K  ...  \
2   212830.94   263546.68   264045.85   264524.14   238057.53  ...
3  2289895.37  2543377.88  2356912.97  2225153.78  1910237.35  ...
0    91723.46   114027.59   114189.36   114155.31   102473.86  ...
4  2376171.92  2631848.13  2433096.60  2292444.02  1964602.88  ...
1   121107.48   149519.09   149856.49   150368.83   135583.68  ...

     100-120K   120-140K   140-160K   160-180K  180-200K     200K+  \
2   184497.42   98567.03   41577.30   26315.88  13533.77   8413.42
3  1033986.58  520388.06  208388.22  125918.09  61405.86  34898.72
0    79199.45   42873.68   18390.53   11864.97   6261.57   4099.84
4  1047576.28  526024.80  210271.20  126881.98  61794.90  35066.78
1   105297.98   55693.35   23186.77   14450.92   7272.20   4313.58

    Mean Income  Median Income  Mode Income  Lower Quartile
```

```
2  52310.696689   43775.286722        27500    25191.421093
3  45098.147714   36701.391428        17500    20343.567416
0  52434.405930   43681.329565        22500    25139.203774
4  44844.080876   36440.438408        17500    20193.027930
1  52217.635985   43845.678743        27500    25231.062847

[5 rows x 33 columns]
```

London Ambulance Data

```
LonAm201820 = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/London Ambulance Service -
Westminster from Police Neighbourhood boundaries 2018-2022/2018-
2020.csv')
LonAm201820.info()
```

```
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/IPython/
core/interactiveshell.py:3444: DtypeWarning:Columns (27) have mixed
types.Specify dtype option on import or set low_memory=False.

<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 168460 entries, 0 to 168459
Data columns (total 44 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   Date                          168460 non-null  object
 1   Time                          168460 non-null  object
 2   Easting                       0 non-null       float64
 3   Northing                      0 non-null       float64
 4   OAs                           168460 non-null  object
 5   LSOA                          168460 non-null  object
 6   LSOA Name                     168460 non-null  object
 7   Ward                          168460 non-null  object
 8   Ward Name                     168460 non-null  object
 9   MPS SNT                       168460 non-null  object
 10  MPS SNT Name                  168460 non-null  object
 11  MPS NHD                       168460 non-null  object
 12  MPS NHD Name                  168460 non-null  object
 13  Borough                       168460 non-null  object
 14  Borough Name                  168460 non-null  object
 15  Police BCU (pre 2018)         168460 non-null  object
 16  Police BCU (pre 2018) Name    168460 non-null  object
 17  Police BCU Sector             168460 non-null  object
 18  Police BCU Sector Name        168460 non-null  object
 19  Police BCU (post 2018)        168460 non-null  object
 20  Police BCU (post 2018) Name   168460 non-null  object
 21  Age                           168460 non-null  int64
 22  Age group                     168460 non-null  object
 23  Age group (youth)             31426 non-null   object
 24  Alcohol related               9329 non-null    object
 25  Ampds determinant             168439 non-null  object
 26  Assault injury              4314 non-null    object
 27  Vehicle id                  11980 non-null   object
 28  Chief complaint             168181 non-null  object
 29  Class a related             252 non-null     object
 30  Destination hospital        168452 non-null  object
 31  Gun injury                  451 non-null     object
 32  Illness                     168460 non-null  object
 33  Incidentid                  168460 non-null  int64
 34  Incident                    168460 non-null  object
 35  Knife injury                851 non-null     object
 36  Overdose                    7101 non-null    object
 37  Self-harm injury            4327 non-null    object
 38  Sex                         165220 non-null  object
 39  Sex injury                  88 non-null      object
 40  ~safestats theme1           14220 non-null   object
 41  ~safestats theme2           1904 non-null    object
 42  ~safestats crime category   4475 non-null    object
 43  ~ SafeStats Crime SubGroup  4475 non-null    object
dtypes: float64(2), int64(2), object(40)
memory usage: 56.6+ MB
```

```
LonAm201820.sample(5, random_state=10)
```

```
              Date      Time  Easting  Northing        OAs       LSOA  
\
141752  2020-07-08  17:13:36      NaN       NaN  E00024001  E01004747   

137718  2020-06-02  01:31:41      NaN       NaN  E00175198  E01004698   

45086   2018-10-20  21:34:05      NaN       NaN  E00023570  E01033603   

156586  2020-11-01  10:54:30      NaN       NaN  E00023922  E01004730   

119502  2020-01-12  07:32:43      NaN       NaN  E00024048  E01004750   


               LSOA Name       Ward       Ward Name  MPS SNT  ...  \
141752  Westminster 021E  E05000646  Vincent Square  00BK19N  ...   
137718  Westminster 017E  E05000638  Lancaster Gate   00BKGJ  ...   
45086   Westminster 009I  E05000634   Church Street  00BK05N  ...   
156586  Westminster 001D  E05000643   Regent's Park     RPRE  ...   
119502  Westminster 023E  E05000647         Warwick  00BK18N  ...   

               Incident Knife injury           Overdose Self-harm 
injury  \
141752            _none          NaN     Caller Derived              
NaN   
137718             Fall          NaN                NaN              
NaN   
45086   Illness (known)          NaN  Paramedic Derived              
```

```
NaN
156586             Fall          NaN                NaN
NaN
119502  Illness (known)          NaN                NaN
NaN

         Sex Sex injury ~safestats theme1 ~safestats theme2  \
141752  Male        NaN               NaN               NaN
137718  Male        NaN               NaN               NaN
45086   Male        NaN               NaN               NaN
156586  Male        NaN               NaN               NaN
119502  Male        NaN               NaN               NaN

       ~safestats crime category ~ SafeStats Crime SubGroup
141752                       NaN                        NaN
137718                       NaN                        NaN
45086                        NaN                        NaN
156586                       NaN                        NaN
119502                       NaN                        NaN

[5 rows x 44 columns]
```

```
LonAm202122 = pd.read_csv('/Users/elika_sinha/Documents/UCL/11. Dissertation/Term3/Datasets/from_WCC/London Ambulance Service - Westminster from Police Neighbourhood boundaries 2018-2022/2021-2022.csv')
LonAm202122.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 77574 entries, 0 to 77573
Data columns (total 44 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Date                          77574 non-null  object
 1   Time                          77574 non-null  object
 2   Easting                       0 non-null      float64
 3   Northing                      0 non-null      float64
 4   OAs                           77574 non-null  object
 5   LSOA                          77574 non-null  object
 6   LSOA Name                     77574 non-null  object
 7   Ward                          77574 non-null  object
 8   Ward Name                     77574 non-null  object
 9   MPS SNT                       77574 non-null  object
 10  MPS SNT Name                  77574 non-null  object
 11  MPS NHD                       77574 non-null  object
 12  MPS NHD Name                  77574 non-null  object
 13  Borough                       77574 non-null  object
 14  Borough Name                  77574 non-null  object
 15  Police BCU (pre 2018)         77574 non-null  object
 16  Police BCU (pre 2018) Name    77574 non-null  object
 17  Police BCU Sector            77574 non-null  object 
 18  Police BCU Sector Name       77574 non-null  object 
 19  Police BCU (post 2018)       77574 non-null  object 
 20  Police BCU (post 2018) Name  77574 non-null  object 
 21  Age                          77574 non-null  int64  
 22  Age group                    77574 non-null  object 
 23  Age group (youth)            14869 non-null  object 
 24  Alcohol related              4558 non-null   object 
 25  Ampds determinant            77510 non-null  object 
 26  Assault injury               1857 non-null   object 
 27  Vehicle id                   77574 non-null  object 
 28  Chief complaint              73538 non-null  object 
 29  Class a related              120 non-null    object 
 30  Destination hospital         77529 non-null  object 
 31  Gun injury                   123 non-null    object 
 32  Illness                      73520 non-null  object 
 33  Incidentid                   77574 non-null  int64  
 34  Incident                     74481 non-null  object 
 35  Knife injury                 319 non-null    object 
 36  Overdose                     2966 non-null   object 
 37  Self-harm injury             1594 non-null   object 
 38  Sex                          75750 non-null  object 
 39  Sex injury                   44 non-null     object 
 40  ~safestats theme1            6684 non-null   object 
 41  ~safestats theme2            802 non-null    object 
 42  ~safestats crime category    1898 non-null   object 
 43  ~ SafeStats Crime SubGroup   1898 non-null   object 
dtypes: float64(2), int64(2), object(40)
memory usage: 26.0+ MB
```

```
LonAm202122.sample(5, random_state=10)
```

```
             Date      Time  Easting  Northing        OAs  
LSOA  \
54365  2021-12-08  12:25:03      NaN       NaN  E00023591  E01004677   

2072   2021-01-11  23:18:31      NaN       NaN  E00023774  E01004710   

58291  2022-01-01  02:47:47      NaN       NaN  E00023930  E01004735   

12780  2021-03-26  04:36:23      NaN       NaN  E00175244  E01004701   

33503  2021-07-31  18:02:56      NaN       NaN  E00023455  E01033607   


              LSOA Name       Ward      Ward Name  MPS SNT  ...  \
54365  Westminster 005C  E05000635    Harrow Road   00BKGF  ...   
2072   Westminster 002D  E05000640     Maida Vale   00BKGL  ...   
58291  Westminster 018B  E05000644    St. James's  00BK07N  ...   
12780  Westminster 007A  E05000639  Little Venice  00BK11N  ...   
```

```
33503  Westminster 014G  E05000631      Bayswater   00BKGB  ...

                Incident Knife injury Overdose Self-harm injury
Sex  \
54365  Illness (unknown)          NaN      NaN              NaN
Female
2072   Illness (unknown)          NaN      NaN              NaN
Male
58291  Illness (unknown)          NaN      NaN              NaN
Unknown
12780  Illness (unknown)          NaN      NaN              NaN
Female
33503  Illness (unknown)          NaN      NaN              NaN
Male

      Sex injury ~safestats theme1 ~safestats theme2  \
54365        NaN               NaN               NaN
2072         NaN               NaN               NaN
58291        NaN   Alcohol-Related               NaN
12780        NaN               NaN               NaN
33503        NaN               NaN               NaN

      ~safestats crime category ~ SafeStats Crime SubGroup
54365                       NaN                        NaN
2072                        NaN                        NaN
58291                       NaN                        NaN
12780                       NaN                        NaN
33503                       NaN                        NaN

[5 rows x 44 columns]
```

Crime Related data

All data is continuous for the time period between July 2018 and July 2022

According to themes

1. Alcohol Related
2. Antisocial Behaviour
3. Fatalities
4. Incidents against emergency services
5. Personal Safety
6. Substance Related
7. Transport Related
8. Weapons
9. Youth

to note- acquisitive offending was 200,000+ and ergo could not be downloaded from safestats

According to crime groups

1. Arson and criminal damage related
2. Burglary related
3. Disorder related
4. Drug related
5. Fraud related
6. Robbery related
7. Violence related
8. Weapon possesion related

to note- theft related was 200,000+ and ergo could not be downloaded from safestats

```
AlcoholRelated = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1339_alcohol
related/safestats_theme_03072022_1339.csv')
AlcoholRelated.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 38714 entries, 0 to 38713
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Dataset                       38714 non-null  object
 1   ~ SafeStats Theme             38714 non-null  object
 2   ~ SafeStats Theme 2           8891 non-null   object
 3   Date                          38714 non-null  object
 4   Time                          14317 non-null  object
 5   Easting                       656 non-null    float64
 6   Northing                      656 non-null    float64
 7   OAs                           14317 non-null  object
 8   LSOA                          14317 non-null  object
 9   LSOA Name                     14317 non-null  object
 10  Ward                          14317 non-null  object
 11  Ward Name                     14317 non-null  object
 12  MPS SNT                       14317 non-null  object
 13  MPS SNT Name                  14317 non-null  object
 14  MPS NHD                       14317 non-null  object
 15  MPS NHD Name                  14317 non-null  object
 16  Borough                       38714 non-null  object
 17  Borough Name                  38714 non-null  object
 18  Police BCU (pre 2018)         38714 non-null  object
 19  Police BCU (pre 2018) Name    38714 non-null  object
 20  Police BCU Sector             38714 non-null  object
 21  Police BCU Sector Name        38714 non-null  object
 22  Police BCU (post 2018)        38714 non-null  object
 23  Police BCU (post 2018) Name   38714 non-null  object 
 24  LAS Incident ID               13661 non-null  float64
 25  ED Incident ID                0 non-null      float64
 26  ED Incident Confidence Score  0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 8.0+ MB
```

```
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/IPython/
core/interactiveshell.py:3444: DtypeWarning:Columns
(4,7,8,9,10,11,12,13,14,15) have mixed types.Specify dtype option on
import or set low_memory=False.
```

```
AntisocialBehaviour =
pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1340_antisocia
l behaviour/safestats_theme_03072022_1340.csv')
AntisocialBehaviour.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 104942 entries, 0 to 104941
Data columns (total 27 columns):
 #   Column                        Non-Null Count   Dtype  
---  ------                        --------------   -----  
 0   Dataset                       104942 non-null  object 
 1   ~ SafeStats Theme             104942 non-null  object 
 2   ~ SafeStats Theme 2           3538 non-null    object 
 3   Date                          104942 non-null  object 
 4   Time                          104942 non-null  object 
 5   Easting                       104457 non-null  float64
 6   Northing                      104457 non-null  float64
 7   OAs                           104457 non-null  object 
 8   LSOA                          104457 non-null  object 
 9   LSOA Name                     104457 non-null  object 
 10  Ward                          104492 non-null  object 
 11  Ward Name                     104492 non-null  object 
 12  MPS SNT                       104492 non-null  object 
 13  MPS SNT Name                  104492 non-null  object 
 14  MPS NHD                       104492 non-null  object 
 15  MPS NHD Name                  104492 non-null  object 
 16  Borough                       104942 non-null  object 
 17  Borough Name                  104942 non-null  object 
 18  Police BCU (pre 2018)         91848 non-null   object 
 19  Police BCU (pre 2018) Name    91848 non-null   object 
 20  Police BCU Sector             104942 non-null  object 
 21  Police BCU Sector Name        104942 non-null  object 
 22  Police BCU (post 2018)        104942 non-null  object 
 23  Police BCU (post 2018) Name   104942 non-null  object 
 24  LAS Incident ID               0 non-null       float64
 25  ED Incident ID                0 non-null       float64
 26  ED Incident Confidence Score  0 non-null       float64
```

```
dtypes: float64(5), object(22)
memory usage: 21.6+ MB
```

```
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/IPython/
core/interactiveshell.py:3444: DtypeWarning:Columns (2,18,19) have
mixed types.Specify dtype option on import or set low_memory=False.
```

```
Fatalities = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1341_fatalitie
s/safestats_theme_03072022_1341.csv')
Fatalities.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 452 entries, 0 to 451
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Dataset                       452 non-null    object
 1   ~ SafeStats Theme             452 non-null    object
 2   ~ SafeStats Theme 2           11 non-null     object
 3   Date                          452 non-null    object
 4   Time                          452 non-null    object
 5   Easting                       10 non-null     float64
 6   Northing                      10 non-null     float64
 7   OAs                           451 non-null    object
 8   LSOA                          451 non-null    object
 9   LSOA Name                     451 non-null    object
 10  Ward                          452 non-null    object
 11  Ward Name                     452 non-null    object
 12  MPS SNT                       452 non-null    object
 13  MPS SNT Name                  452 non-null    object
 14  MPS NHD                       452 non-null    object
 15  MPS NHD Name                  452 non-null    object
 16  Borough                       452 non-null    object
 17  Borough Name                  452 non-null    object
 18  Police BCU (pre 2018)         442 non-null    object
 19  Police BCU (pre 2018) Name    442 non-null    object
 20  Police BCU Sector             452 non-null    object
 21  Police BCU Sector Name        452 non-null    object
 22  Police BCU (post 2018)        452 non-null    object
 23  Police BCU (post 2018) Name   452 non-null    object
 24  LAS Incident ID               441 non-null    float64
 25  ED Incident ID                0 non-null      float64
 26  ED Incident Confidence Score  0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 95.5+ KB
```

```
EmergencyServices = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1342_incidents
against emergency services/safestats_theme_03072022_1342.csv')
EmergencyServices.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 1625 entries, 0 to 1624
Data columns (total 27 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Dataset                         1625 non-null   object 
 1   ~ SafeStats Theme               407 non-null    object 
 2   ~ SafeStats Theme 2             1625 non-null   object 
 3   Date                            1625 non-null   object 
 4   Time                            1625 non-null   object 
 5   Easting                         215 non-null    float64
 6   Northing                        215 non-null    float64
 7   OAs                             1625 non-null   object 
 8   LSOA                            1625 non-null   object 
 9   LSOA Name                       1625 non-null   object 
 10  Ward                            1625 non-null   object 
 11  Ward Name                       1625 non-null   object 
 12  MPS SNT                         1625 non-null   object 
 13  MPS SNT Name                    1625 non-null   object 
 14  MPS NHD                         1625 non-null   object 
 15  MPS NHD Name                    1625 non-null   object 
 16  Borough                         1625 non-null   object 
 17  Borough Name                    1625 non-null   object 
 18  Police BCU (pre 2018)           1625 non-null   object 
 19  Police BCU (pre 2018) Name      1625 non-null   object 
 20  Police BCU Sector               1625 non-null   object 
 21  Police BCU Sector Name          1625 non-null   object 
 22  Police BCU (post 2018)          1625 non-null   object 
 23  Police BCU (post 2018) Name     1625 non-null   object 
 24  LAS Incident ID                 1410 non-null   float64
 25  ED Incident ID                  0 non-null      float64
 26  ED Incident Confidence Score    0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 342.9+ KB
```

```
PersonalSafety = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1343_personal
safety/safestats_theme_03072022_1343.csv')
PersonalSafety.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 113258 entries, 0 to 113257
Data columns (total 27 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   Dataset                         113258 non-null  object 
 1   ~ SafeStats Theme               71097 non-null   object 
 2   ~ SafeStats Theme 2             56015 non-null   object 
 3   Date                            113258 non-null  object 
 4   Time                            113258 non-null  object 
 5   Easting                         102720 non-null  float64
 6   Northing                      102720 non-null  float64
 7   OAs                           108536 non-null  object
 8   LSOA                          108536 non-null  object
 9   LSOA Name                     108536 non-null  object
 10  Ward                          111180 non-null  object
 11  Ward Name                     111180 non-null  object
 12  MPS SNT                       111180 non-null  object
 13  MPS SNT Name                  111180 non-null  object
 14  MPS NHD                       111180 non-null  object
 15  MPS NHD Name                  111180 non-null  object
 16  Borough                       113258 non-null  object
 17  Borough Name                  113258 non-null  object
 18  Police BCU (pre 2018)         60430 non-null   object
 19  Police BCU (pre 2018) Name    60430 non-null   object
 20  Police BCU Sector             113258 non-null  object
 21  Police BCU Sector Name        113258 non-null  object
 22  Police BCU (post 2018)        113258 non-null  object
 23  Police BCU (post 2018) Name   113258 non-null  object
 24  LAS Incident ID               5816 non-null    float64
 25  ED Incident ID                633 non-null     object
 26  ED Incident Confidence Score  633 non-null     float64
dtypes: float64(4), object(23)
memory usage: 23.3+ MB
```

```
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/IPython/
core/interactiveshell.py:3444: DtypeWarning:Columns (18,19,25) have
mixed types.Specify dtype option on import or set low_memory=False.
```

```
PersonalSafety = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1343_personal
safety/safestats_theme_03072022_1343.csv')
PersonalSafety.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 113258 entries, 0 to 113257
Data columns (total 27 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   Dataset                       113258 non-null  object
 1   ~ SafeStats Theme             71097 non-null   object
 2   ~ SafeStats Theme 2           56015 non-null   object
 3   Date                          113258 non-null  object
 4   Time                          113258 non-null  object
 5   Easting                       102720 non-null  float64
 6   Northing                      102720 non-null  float64
 7   OAs                           108536 non-null  object
 8   LSOA                          108536 non-null  object
 9   LSOA Name                     108536 non-null  object
 10  Ward                          111180 non-null  object
 11  Ward Name                     111180 non-null  object
 12  MPS SNT                       111180 non-null  object
 13  MPS SNT Name                     111180 non-null  object 
 14  MPS NHD                          111180 non-null  object 
 15  MPS NHD Name                     111180 non-null  object 
 16  Borough                          113258 non-null  object 
 17  Borough Name                     113258 non-null  object 
 18  Police BCU (pre 2018)            60430 non-null   object 
 19  Police BCU (pre 2018) Name       60430 non-null   object 
 20  Police BCU Sector                113258 non-null  object 
 21  Police BCU Sector Name           113258 non-null  object 
 22  Police BCU (post 2018)           113258 non-null  object 
 23  Police BCU (post 2018) Name      113258 non-null  object 
 24  LAS Incident ID                  5816 non-null    float64
 25  ED Incident ID                   633 non-null     object 
 26  ED Incident Confidence Score     633 non-null     float64
dtypes: float64(4), object(23)
memory usage: 23.3+ MB
```

```
SubstanceRelated = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1345_substance
related/safestats_theme_03072022_1345.csv')
SubstanceRelated.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 55129 entries, 0 to 55128
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Dataset                       55129 non-null  object 
 1   ~ SafeStats Theme             27457 non-null  object 
 2   ~ SafeStats Theme 2           35810 non-null  object 
 3   Date                          55129 non-null  object 
 4   Time                          19370 non-null  object 
 5   Easting                       18665 non-null  float64
 6   Northing                      18665 non-null  float64
 7   OAs                           19062 non-null  object 
 8   LSOA                          19062 non-null  object 
 9   LSOA Name                     19062 non-null  object 
 10  Ward                          19135 non-null  object 
 11  Ward Name                     19135 non-null  object 
 12  MPS SNT                       19135 non-null  object 
 13  MPS SNT Name                  19135 non-null  object 
 14  MPS NHD                       19135 non-null  object 
 15  MPS NHD Name                  19135 non-null  object 
 16  Borough                       55129 non-null  object 
 17  Borough Name                  55129 non-null  object 
 18  Police BCU (pre 2018)         47477 non-null  object 
 19  Police BCU (pre 2018) Name    47477 non-null  object 
 20  Police BCU Sector             55129 non-null  object 
 21  Police BCU Sector Name        55129 non-null  object 
 22  Police BCU (post 2018)        55129 non-null  object 
 23  Police BCU (post 2018) Name   55129 non-null  object 
 24  LAS Incident ID                397 non-null    float64
 25  ED Incident ID                 0 non-null      float64
 26  ED Incident Confidence Score   0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 11.4+ MB
```

```
TransportRelated = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1346_transport
related/safestats_theme_03072022_1346.csv')
TransportRelated.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 38799 entries, 0 to 38798
Data columns (total 27 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Dataset                        38799 non-null  object
 1   ~safestats theme1              33395 non-null  object
 2   ~safestats theme2              35673 non-null  object
 3   Date                           38799 non-null  object
 4   Time                           38799 non-null  object
 5   Easting                        38291 non-null  float64
 6   Northing                       38291 non-null  float64
 7   OAs                            38494 non-null  object
 8   LSOA                           38494 non-null  object
 9   LSOA Name                      38494 non-null  object
 10  Ward                           38501 non-null  object
 11  Ward Name                      38501 non-null  object
 12  MPS SNT                        38501 non-null  object
 13  MPS SNT Name                   38501 non-null  object
 14  MPS NHD                        38501 non-null  object
 15  MPS NHD Name                   38501 non-null  object
 16  Borough                        38799 non-null  object
 17  Borough Name                   38799 non-null  object
 18  Police BCU (pre 2018)          26283 non-null  object
 19  Police BCU (pre 2018) Name     26283 non-null  object
 20  Police BCU Sector              38799 non-null  object
 21  Police BCU Sector Name         38799 non-null  object
 22  Police BCU (post 2018)         38799 non-null  object
 23  Police BCU (post 2018) Name    38799 non-null  object
 24  LAS Incident ID                203 non-null    float64
 25  ED Incident ID                 0 non-null      float64
 26  ED Incident Confidence Score   0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 8.0+ MB
```

```
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/IPython/
core/interactiveshell.py:3444: DtypeWarning:Columns (18,19) have mixed
types.Specify dtype option on import or set low_memory=False.
```

```
Weapons = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1354_weapons/
safestats_theme_03072022_1354.csv')
Weapons.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 3561 entries, 0 to 3560
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Dataset                       3561 non-null   object 
 1   ~ SafeStats Theme             2275 non-null   object 
 2   ~ SafeStats Theme 2           3557 non-null   object 
 3   Date                          3561 non-null   object 
 4   Time                          3561 non-null   object 
 5   Easting                       2342 non-null   float64
 6   Northing                      2342 non-null   float64
 7   OAs                           3522 non-null   object 
 8   LSOA                          3522 non-null   object 
 9   LSOA Name                     3522 non-null   object 
 10  Ward                          3537 non-null   object 
 11  Ward Name                     3537 non-null   object 
 12  MPS SNT                       3537 non-null   object 
 13  MPS SNT Name                  3537 non-null   object 
 14  MPS NHD                       3537 non-null   object 
 15  MPS NHD Name                  3537 non-null   object 
 16  Borough                       3561 non-null   object 
 17  Borough Name                  3561 non-null   object 
 18  Police BCU (pre 2018)         2785 non-null   object 
 19  Police BCU (pre 2018) Name    2785 non-null   object 
 20  Police BCU Sector             3561 non-null   object 
 21  Police BCU Sector Name        3561 non-null   object 
 22  Police BCU (post 2018)        3561 non-null   object 
 23  Police BCU (post 2018) Name   3561 non-null   object 
 24  LAS Incident ID               1180 non-null   float64
 25  ED Incident ID                162 non-null    object 
 26  ED Incident Confidence Score  162 non-null    float64
dtypes: float64(4), object(23)
memory usage: 751.3+ KB
```

```
Youth = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1354_youth/
safestats_theme_03072022_1354.csv')
Youth.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 2118 entries, 0 to 2117
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Dataset                       2118 non-null   object 
 1   ~ SafeStats Theme             2118 non-null   object 
 2   ~ SafeStats Theme 2           0 non-null      float64
 3   Date                          2118 non-null   object 
 4   Time                          0 non-null      float64
 5   Easting                       0 non-null      float64
 6   Northing                      0 non-null      float64
 7   OAs                           0 non-null      float64
 8   LSOA                          0 non-null      float64
 9   LSOA Name                     0 non-null      float64
 10  Ward                          0 non-null      float64
 11  Ward Name                     0 non-null      float64
 12  MPS SNT                       0 non-null      float64
 13  MPS SNT Name                  0 non-null      float64
 14  MPS NHD                       0 non-null      float64
 15  MPS NHD Name                  0 non-null      float64
 16  Borough                       2118 non-null   object 
 17  Borough Name                  2118 non-null   object 
 18  Police BCU (pre 2018)         2118 non-null   object 
 19  Police BCU (pre 2018) Name    2118 non-null   object 
 20  Police BCU Sector             2118 non-null   object 
 21  Police BCU Sector Name        2118 non-null   object 
 22  Police BCU (post 2018)        2118 non-null   object 
 23  Police BCU (post 2018) Name   2118 non-null   object 
 24  LAS Incident ID               0 non-null      float64
 25  ED Incident ID                0 non-null      float64
 26  ED Incident Confidence Score  0 non-null      float64
dtypes: float64(16), object(11)
memory usage: 446.9+ KB
```

```
Damage = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1356_arson and
criminal damage related/safestats_crime_group_03072022_1356.csv')
Damage.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 14077 entries, 0 to 14076
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Dataset                       14077 non-null  object 
 1   ~ SafeStats Crime Group       14077 non-null  object 
 2   ~ SafeStats Crime SubGroup    14077 non-null  object 
 3   Date                          14077 non-null  object 
 4   Time                          14077 non-null  object 
 5   Easting                       13981 non-null  float64
 6   Northing                      13981 non-null  float64
 7   OAs                           13981 non-null  object 
 8   LSOA                          13981 non-null  object 
 9   LSOA Name                     13981 non-null  object 
 10  Ward                          13991 non-null  object 
 11  Ward Name                     13991 non-null  object 
 12  MPS SNT                       13991 non-null  object 
 13  MPS SNT Name                  13991 non-null  object 
 14  MPS NHD                       13991 non-null  object 
 15  MPS NHD Name                  13991 non-null  object 
 16  Borough                       14077 non-null  object 
 17  Borough Name                  14077 non-null  object 
 18  Police BCU (pre 2018)         8780 non-null   object 
 19  Police BCU (pre 2018) Name    8780 non-null   object 
 20  Police BCU Sector             14077 non-null  object 
 21  Police BCU Sector Name        14077 non-null  object 
 22  Police BCU (post 2018)        14077 non-null  object 
 23  Police BCU (post 2018) Name   14077 non-null  object 
 24  LAS Incident ID               0 non-null      float64
 25  ED Incident ID                0 non-null      float64
 26  ED Incident Confidence Score  0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 2.9+ MB
```

```
Burglary = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1405_burglary
related/safestats_crime_group_03072022_1405.csv')
Burglary.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 19994 entries, 0 to 19993
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Dataset                       19994 non-null  object 
 1   ~ SafeStats Crime Group       19994 non-null  object 
 2   ~ SafeStats Crime SubGroup    19994 non-null  object 
 3   Date                          19994 non-null  object 
 4   Time                          19994 non-null  object 
 5   Easting                       19940 non-null  float64
 6   Northing                      19940 non-null  float64
 7   OAs                           19940 non-null  object 
 8   LSOA                          19940 non-null  object 
 9   LSOA Name                     19940 non-null  object 
 10  Ward                          19948 non-null  object 
 11  Ward Name                     19948 non-null  object 
 12  MPS SNT                       19948 non-null  object 
 13  MPS SNT Name                  19948 non-null  object 
 14  MPS NHD                       19948 non-null  object 
 15  MPS NHD Name                  19948 non-null  object 
 16  Borough                       19994 non-null  object 
 17  Borough Name                  19994 non-null  object 
 18  Police BCU (pre 2018)         13546 non-null  object 
 19  Police BCU (pre 2018) Name    13546 non-null  object 
 20  Police BCU Sector             19994 non-null  object 
 21  Police BCU Sector Name        19994 non-null  object 
 22  Police BCU (post 2018)        19994 non-null  object 
 23  Police BCU (post 2018) Name   19994 non-null  object
 24  LAS Incident ID               0 non-null      float64
 25  ED Incident ID                0 non-null      float64
 26  ED Incident Confidence Score  0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 4.1+ MB
```

```
Disorder = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1405_disorder
related/safestats_crime_group_03072022_1405.csv')
Disorder.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 78977 entries, 0 to 78976
Data columns (total 27 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Dataset                       78977 non-null  object
 1   ~ SafeStats Crime Group       78977 non-null  object
 2   ~ SafeStats Crime SubGroup    78977 non-null  object
 3   Date                          78977 non-null  object
 4   Time                          78977 non-null  object
 5   Easting                       78584 non-null  float64
 6   Northing                      78584 non-null  float64
 7   OAs                           78584 non-null  object
 8   LSOA                          78584 non-null  object
 9   LSOA Name                     78584 non-null  object
 10  Ward                          78610 non-null  object
 11  Ward Name                     78610 non-null  object
 12  MPS SNT                       78610 non-null  object
 13  MPS SNT Name                  78610 non-null  object
 14  MPS NHD                       78610 non-null  object
 15  MPS NHD Name                  78610 non-null  object
 16  Borough                       78977 non-null  object
 17  Borough Name                  78977 non-null  object
 18  Police BCU (pre 2018)         71036 non-null  object
 19  Police BCU (pre 2018) Name    71036 non-null  object
 20  Police BCU Sector             78977 non-null  object
 21  Police BCU Sector Name        78977 non-null  object
 22  Police BCU (post 2018)        78977 non-null  object
 23  Police BCU (post 2018) Name   78977 non-null  object
 24  LAS Incident ID               0 non-null      float64
 25  ED Incident ID                0 non-null      float64
 26  ED Incident Confidence Score  0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 16.3+ MB
```

```
Fraud = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1408_fraud
related/safestats_crime_group_03072022_1408.csv')
Fraud.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 1960 entries, 0 to 1959
Data columns (total 27 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Dataset                         1960 non-null   object 
 1   ~ SafeStats Crime Group         1960 non-null   object 
 2   ~ SafeStats Crime SubGroup      1960 non-null   object 
 3   Date                            1960 non-null   object 
 4   Time                            1960 non-null   object 
 5   Easting                         1952 non-null   float64
 6   Northing                        1952 non-null   float64
 7   OAs                             1952 non-null   object 
 8   LSOA                            1952 non-null   object 
 9   LSOA Name                       1952 non-null   object 
 10  Ward                            1952 non-null   object 
 11  Ward Name                       1952 non-null   object 
 12  MPS SNT                         1952 non-null   object 
 13  MPS SNT Name                    1952 non-null   object 
 14  MPS NHD                         1952 non-null   object 
 15  MPS NHD Name                    1952 non-null   object 
 16  Borough                         1960 non-null   object 
 17  Borough Name                    1960 non-null   object 
 18  Police BCU (pre 2018)           1769 non-null   object 
 19  Police BCU (pre 2018) Name      1769 non-null   object 
 20  Police BCU Sector               1960 non-null   object 
 21  Police BCU Sector Name          1960 non-null   object 
 22  Police BCU (post 2018)          1960 non-null   object 
 23  Police BCU (post 2018) Name     1960 non-null   object 
 24  LAS Incident ID                 0 non-null      float64
 25  ED Incident ID                  0 non-null      float64
 26  ED Incident Confidence Score    0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 413.6+ KB
```

```
Robbery = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1408_robbery
related/safestats_crime_group_03072022_1408.csv')
Robbery.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 16828 entries, 0 to 16827
Data columns (total 27 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Dataset                         16828 non-null  object 
 1   ~ SafeStats Crime Group         16828 non-null  object 
 2   ~ SafeStats Crime SubGroup      16828 non-null  object 
 3   Date                            16828 non-null  object 
 4   Time                            16828 non-null  object 
 5   Easting                         16670 non-null  float64
 6   Northing                       16670 non-null  float64
 7   OAs                            16670 non-null  object
 8   LSOA                           16670 non-null  object
 9   LSOA Name                      16670 non-null  object
 10  Ward                           16677 non-null  object
 11  Ward Name                      16677 non-null  object
 12  MPS SNT                        16677 non-null  object
 13  MPS SNT Name                   16677 non-null  object
 14  MPS NHD                        16677 non-null  object
 15  MPS NHD Name                   16677 non-null  object
 16  Borough                        16828 non-null  object
 17  Borough Name                   16828 non-null  object
 18  Police BCU (pre 2018)          11311 non-null  object
 19  Police BCU (pre 2018) Name     11311 non-null  object
 20  Police BCU Sector              16828 non-null  object
 21  Police BCU Sector Name         16828 non-null  object
 22  Police BCU (post 2018)         16828 non-null  object
 23  Police BCU (post 2018) Name    16828 non-null  object
 24  LAS Incident ID                0 non-null      float64
 25  ED Incident ID                 0 non-null      float64
 26  ED Incident Confidence Score   0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 3.5+ MB
```

```
SexRelated = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1409_sex
related/safestats_crime_group_03072022_1409.csv')
SexRelated.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 3548 entries, 0 to 3547
Data columns (total 27 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Dataset                        3548 non-null   object
 1   ~ SafeStats Crime Group        3548 non-null   object
 2   ~ SafeStats Crime SubGroup     3548 non-null   object
 3   Date                           3548 non-null   object
 4   Time                           3548 non-null   object
 5   Easting                        723 non-null    float64
 6   Northing                       723 non-null    float64
 7   OAs                            843 non-null    object
 8   LSOA                           843 non-null    object
 9   LSOA Name                      843 non-null    object
 10  Ward                           3224 non-null   object
 11  Ward Name                      3224 non-null   object
 12  MPS SNT                        3224 non-null   object
 13  MPS SNT Name                   3224 non-null   object
 14  MPS NHD                        3224 non-null   object
 15  MPS NHD Name                   3224 non-null   object
 16  Borough                        3548 non-null   object
 17  Borough Name                   3548 non-null   object 
 18  Police BCU (pre 2018)          843 non-null    object 
 19  Police BCU (pre 2018) Name     843 non-null    object 
 20  Police BCU Sector              3548 non-null   object 
 21  Police BCU Sector Name         3548 non-null   object 
 22  Police BCU (post 2018)         3548 non-null   object 
 23  Police BCU (post 2018) Name    3548 non-null   object 
 24  LAS Incident ID                120 non-null    float64
 25  ED Incident ID                 0 non-null      float64
 26  ED Incident Confidence Score   0 non-null      float64
dtypes: float64(5), object(22)
memory usage: 748.5+ KB
```

```
Violence = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1410_violence
related/safestats_crime_group_03072022_1410.csv')
Violence.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 71614 entries, 0 to 71613
Data columns (total 27 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Dataset                        71614 non-null  object 
 1   ~ SafeStats Crime Group        71614 non-null  object 
 2   ~ SafeStats Crime SubGroup     71614 non-null  object 
 3   Date                           71614 non-null  object 
 4   Time                           71614 non-null  object 
 5   Easting                        64720 non-null  float64
 6   Northing                       64720 non-null  float64
 7   OAs                            70430 non-null  object 
 8   LSOA                           70430 non-null  object 
 9   LSOA Name                      70430 non-null  object 
 10  Ward                           70678 non-null  object 
 11  Ward Name                      70678 non-null  object 
 12  MPS SNT                        70678 non-null  object 
 13  MPS SNT Name                   70678 non-null  object 
 14  MPS NHD                        70678 non-null  object 
 15  MPS NHD Name                   70678 non-null  object 
 16  Borough                        71614 non-null  object 
 17  Borough Name                   71614 non-null  object 
 18  Police BCU (pre 2018)          48182 non-null  object 
 19  Police BCU (pre 2018) Name     48182 non-null  object 
 20  Police BCU Sector              71614 non-null  object 
 21  Police BCU Sector Name         71614 non-null  object 
 22  Police BCU (post 2018)         71614 non-null  object 
 23  Police BCU (post 2018) Name    71614 non-null  object 
 24  LAS Incident ID                5710 non-null   float64
 25  ED Incident ID                 633 non-null    object 
 26  ED Incident Confidence Score   633 non-null    float64
```

```
dtypes: float64(4), object(23)
memory usage: 14.8+ MB
```

```
WeaponPossession = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/safestats_03072022_1411_weapon
posession related/safestats_crime_group_03072022_1411.csv')
Violence.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 71614 entries, 0 to 71613
Data columns (total 27 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Dataset                         71614 non-null  object
 1   ~ SafeStats Crime Group         71614 non-null  object
 2   ~ SafeStats Crime SubGroup      71614 non-null  object
 3   Date                            71614 non-null  object
 4   Time                            71614 non-null  object
 5   Easting                         64720 non-null  float64
 6   Northing                        64720 non-null  float64
 7   OAs                             70430 non-null  object
 8   LSOA                            70430 non-null  object
 9   LSOA Name                       70430 non-null  object
 10  Ward                            70678 non-null  object
 11  Ward Name                       70678 non-null  object
 12  MPS SNT                         70678 non-null  object
 13  MPS SNT Name                    70678 non-null  object
 14  MPS NHD                         70678 non-null  object
 15  MPS NHD Name                    70678 non-null  object
 16  Borough                         71614 non-null  object
 17  Borough Name                    71614 non-null  object
 18  Police BCU (pre 2018)           48182 non-null  object
 19  Police BCU (pre 2018) Name      48182 non-null  object
 20  Police BCU Sector               71614 non-null  object
 21  Police BCU Sector Name          71614 non-null  object
 22  Police BCU (post 2018)          71614 non-null  object
 23  Police BCU (post 2018) Name     71614 non-null  object
 24  LAS Incident ID                 5710 non-null   float64
 25  ED Incident ID                  633 non-null    object
 26  ED Incident Confidence Score    633 non-null    float64
dtypes: float64(4), object(23)
memory usage: 14.8+ MB
```

Noise and Odour

```
Noise_odour = pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/Noise and odour/Noise and
odour.xlsx', header=0)
Noise_odour.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
```

```
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype         
---  ------                          --------------  -----         
 0   Noise Complaint Index           90798 non-null  object        
 1   Time                            90798 non-null  object        
 2   Received Date                   90798 non-null  datetime64[ns]
 3   Financial Year                  90798 non-null  object        
 4   Service Request                 90798 non-null  object        
 5   Service Request Group Type      90798 non-null  object        
 6   Service Request Group Sub Type  90798 non-null  object        
 7   Service Request_1               90798 non-null  object        
 8   Address Key                     90798 non-null  object        
 9   Type of Address                 90798 non-null  object        
 10  Output Area Code (2011)         90798 non-null  object        
 11  MSOA 2011 Code                  90798 non-null  object        
 12  MSOA 2011 Name                  90798 non-null  object        
 13  LSOA 2011 Code                  90798 non-null  object        
 14  LSOA 2011 Name                  90798 non-null  object        
 15  Ward Name                       90798 non-null  object        
dtypes: datetime64[ns](1), object(15)
memory usage: 11.1+ MB
```

Licensing Data

```
Licensing = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/Licensing Data/SR-179077.csv')
Licensing.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 249616 entries, 0 to 249615
Data columns (total 11 columns):
 #   Column    Non-Null Count   Dtype 
---  ------    --------------   ----- 
 0   REFVAL    249616 non-null  object
 1   ADDRESS   249616 non-null  object
 2   LIUSE     237886 non-null  object
 3   ISSUED    249616 non-null  object
 4   LIPERMIT  248905 non-null  object
 5   LICYCLE   224063 non-null  object
 6   FTYPE     249616 non-null  object
 7   FVALUE    232187 non-null  object
 8   OPENT     223901 non-null  object
 9   CLOST     223906 non-null  object
 10  STREET    249616 non-null  object
dtypes: object(11)
memory usage: 20.9+ MB
```

Cleaning datasets

1. Choosing relevant columns across all datasets
2. Merging all relevant columns into one dataset

Paycheck Data from CACI

```
Paycheck22_OA = pd.read_excel('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/CACI Paycheck Data
(Income)/Westminster City Council - Westminster - Paycheck directory
2022.xlsx', sheet_name='OA', header=8)
Paycheck22_OA.info()
Paycheck22_OA.sample(6, random_state=10)
```

```
       Area ID  Area Name  Total households  0-5K  5-10K  10-15K  15-
20K  \
27   E00023437        NaN               110  0.09   0.66    1.73
2.95
305  E00023732        NaN               131  0.13   0.92    2.37
3.98
579  E00024023        NaN               184  0.61   3.64    7.60
10.72
514  E00023949        NaN               136  0.70   4.00    7.99
10.77
181  E00023599        NaN               127  0.76   4.10    7.81
10.21
588  E00024033        NaN               117  0.15   1.02    2.47
3.96

     20-25K  25-30K  30-35K  ...  100-120K  120-140K  140-160K  160-
180K  \
27     3.77    4.57    4.80  ...     11.17      6.91      3.28
2.29
305    5.03    6.04    6.28  ...     12.18      7.11      3.18
2.09
579   11.79   12.65   12.00  ...     10.62      5.41      2.16
1.29
514   11.30   11.56   10.46  ...      4.85      2.13      0.74
0.38
181   10.51   10.65    9.60  ...      4.47      1.91      0.64
0.32
588    4.82    5.62    5.72  ...     10.69      6.41      2.96
2.02

     180-200K  200K+   Mean Income  Median Income  Mode Income  Lower
Quartile
27       1.32   0.99  74502.727273   67158.203125     110000.0
43004.966887
305      1.11   0.69  70203.816794   63118.361153     110000.0
40721.649485
579      0.62   0.34  56000.543478   48666.069830      42500.0
29600.790514
514      0.16   0.06  47063.235294   40568.295115      27500.0
24663.716814
181      0.12   0.05  46676.377953   40325.027086      27500.0
24219.790676
```

```
588      1.12   0.77  69987.179487   62340.989399     110000.0
39456.168831

[6 rows x 33 columns]
```

```
PayCheck = Paycheck22_OA.drop(columns=['Area Name','0-5K', '5-10K',
'10-15K', '15-20K', '20-25K', '25-30K', '30-35K', '35-40K', '40-45K',
'45-50K', '50-55K', '55-60K', '60-65K', '65-70K', '70-75K', '75-80K',
'80-85K', '85-90K', '90-95K', '95-100K', '100-120K', '120-140K', '140-
160K', '160-180K', '180-200K', '200K+'])
print(PayCheck)
PayCheck.sample(6, random_state=10)
```

```
       Area ID  Total households   Mean Income  Median Income  Mode
Income  \
27   E00023437               110  74502.727273   67158.203125
110000.0
305  E00023732               131  70203.816794   63118.361153
110000.0
579  E00024023               184  56000.543478   48666.069830
42500.0
514  E00023949               136  47063.235294   40568.295115
27500.0
181  E00023599               127  46676.377953   40325.027086
27500.0
588  E00024033               117  69987.179487   62340.989399
110000.0

     Lower Quartile
27     43004.966887
305    40721.649485
579    29600.790514
514    24663.716814
181    24219.790676
588    39456.168831
```

```
PayCheck = PayCheck.rename(columns={'Area ID': 'OAs'})
PayCheck.sample(6, random_state=10)
```

```
           OAs  Total households   Mean Income  Median Income  Mode
Income  \
27   E00023437               110  74502.727273   67158.203125
110000.0
305  E00023732               131  70203.816794   63118.361153
110000.0
579  E00024023               184  56000.543478   48666.069830
42500.0
514  E00023949               136  47063.235294   40568.295115
27500.0
181  E00023599               127  46676.377953   40325.027086
27500.0
```

```
588  E00024033               117  69987.179487   62340.989399
110000.0

     Lower Quartile
27     43004.966887
305    40721.649485
579    29600.790514
514    24663.716814
181    24219.790676
588    39456.168831
```

```
PayCheck.to_csv('PayCheck.csv', encoding='utf-8', index=False)
```

London Ambulance Data

```
LonAm201820 = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/London Ambulance Service -
Westminster from Police Neighbourhood boundaries 2018-2022/2018-
2020.csv')
LonAm201820.info()
LonAm201820.sample(6, random_state=10)
```

```
              Date      Time  Easting  Northing        OAs       LSOA
\
141752  2020-07-08  17:13:36      NaN       NaN  E00024001  E01004747

137718  2020-06-02  01:31:41      NaN       NaN  E00175198  E01004698

45086   2018-10-20  21:34:05      NaN       NaN  E00023570  E01033603

156586  2020-11-01  10:54:30      NaN       NaN  E00023922  E01004730

119502  2020-01-12  07:32:43      NaN       NaN  E00024048  E01004750

24692   2018-06-15  17:30:23      NaN       NaN  E00023930  E01004735


               LSOA Name       Ward       Ward Name  MPS SNT  ...  \
141752  Westminster 021E  E05000646  Vincent Square  00BK19N  ...
137718  Westminster 017E  E05000638  Lancaster Gate   00BKGJ  ...
45086   Westminster 009I  E05000634   Church Street  00BK05N  ...
156586  Westminster 001D  E05000643   Regent's Park     RPRE  ...
119502  Westminster 023E  E05000647         Warwick  00BK18N  ...
24692   Westminster 018B  E05000644     St. James's  00BK07N  ...

                 Incident Knife injury           Overdose Self-harm
injury  \
141752              _none          NaN     Caller Derived
NaN
137718               Fall          NaN                NaN
NaN
```

```
45086     Illness (known)          NaN  Paramedic Derived
NaN
156586               Fall          NaN                NaN
NaN
119502    Illness (known)          NaN                NaN
NaN
24692   Illness (unknown)          NaN                NaN
NaN

           Sex Sex injury ~safestats theme1 ~safestats theme2  \
141752    Male        NaN               NaN               NaN
137718    Male        NaN               NaN               NaN
45086     Male        NaN               NaN               NaN
156586    Male        NaN               NaN               NaN
119502    Male        NaN               NaN               NaN
24692   Female        NaN               NaN               NaN

       ~safestats crime category ~ SafeStats Crime SubGroup
141752                       NaN                        NaN
137718                       NaN                        NaN
45086                        NaN                        NaN
156586                       NaN                        NaN
119502                       NaN                        NaN
24692                        NaN                        NaN

[6 rows x 44 columns]
```

```
LonAM1 = LonAm201820.drop(columns=['Date','Time', 'Easting',
'Northing', 'LSOA', 'LSOA Name', 'Ward', 'Ward Name', 'MPS SNT', 'MPS
SNT Name', 'MPS NHD', 'MPS NHD Name', 'Borough', 'Borough Name',
'Police BCU (pre 2018)', 'Police BCU (pre 2018) Name', 'Police BCU
Sector', 'Police BCU Sector Name', 'Police BCU (post 2018)', 'Police
BCU (post 2018) Name', 'Age', 'Age group', 'Age group (youth)', 'Ampds
determinant', 'Vehicle id', 'Chief complaint', 'Illness', 'Alcohol
related', 'Assault injury', 'Class a related', 'Gun injury', 'Knife
injury', 'Overdose', 'Self-harm injury', 'Sex injury', 'Destination
hospital', 'Incidentid', 'Sex', '~safestats theme1', '~safestats
theme2', '~safestats crime category', '~ SafeStats Crime SubGroup'])
print(LonAM1)
LonAM1.sample(33, random_state=10)
```

```
              OAs                    Incident
141752  E00024001                       _none
137718  E00175198                        Fall
45086   E00023570             Illness (known)
156586  E00023922                        Fall
119502  E00024048             Illness (known)
24692   E00023930           Illness (unknown)
1810    E00023710                       _none
149521  E00175271                 Psychiatric
```

```
76749   E00023591              Illness (unknown)
70363   E00175249                Illness (known)
12291   E00024032                          _none
2906    E00023779                       Accident
56797   E00023490              Illness (unknown)
141516  E00023429              Illness (unknown)
154461  E00024048                           Fall
14246   E00023748                Illness (known)
84232   E00023635  Mental Health Act Section 136
36175   E00175206                Illness (known)
39605   E00024051                          _none
98256   E00023580                Illness (known)
144446  E00023714                          _none
12903   E00024104              Illness (unknown)
149597  E00024048                          _none
42151   E00023866                Illness (known)
98684   E00024028              Illness (unknown)
148750  E00023944                Illness (known)
74520   E00175253              RTC - not trapped
148158  E00175186              Hospital Transfer
112857  E00023574                Illness (known)
168194  E00024005                          _none
4820    E00023937              Illness (unknown)
107005  E00023933              Illness (unknown)
168409  E00023798                Illness (known)
```

```
LonAM1 = LonAM1.dropna(axis=1,how='all')
```

```
LonAM1['Incident1820_no'] = 1
LonAM1.sample(33, random_state=10)
```

```
              OAs                       Incident  Incident1820_no
141752  E00024001                          _none                1
137718  E00175198                           Fall                1
45086   E00023570                Illness (known)                1
156586  E00023922                           Fall                1
119502  E00024048                Illness (known)                1
24692   E00023930              Illness (unknown)                1
1810    E00023710                          _none                1
149521  E00175271                    Psychiatric                1
76749   E00023591              Illness (unknown)                1
70363   E00175249                Illness (known)                1
12291   E00024032                          _none                1
2906    E00023779                       Accident                1
56797   E00023490              Illness (unknown)                1
141516  E00023429              Illness (unknown)                1
154461  E00024048                           Fall                1
14246   E00023748                Illness (known)                1
84232   E00023635  Mental Health Act Section 136                1
36175   E00175206                Illness (known)                1
39605   E00024051                          _none                1
```

```
98256   E00023580                Illness (known)                1
144446  E00023714                          _none                1
12903   E00024104              Illness (unknown)                1
149597  E00024048                          _none                1
42151   E00023866                Illness (known)                1
98684   E00024028              Illness (unknown)                1
148750  E00023944                Illness (known)                1
74520   E00175253              RTC - not trapped                1
148158  E00175186              Hospital Transfer                1
112857  E00023574                Illness (known)                1
168194  E00024005                          _none                1
4820    E00023937              Illness (unknown)                1
107005  E00023933              Illness (unknown)                1
168409  E00023798                Illness (known)                1
```

```
LonAm202122 = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/from_WCC/London Ambulance Service -
Westminster from Police Neighbourhood boundaries 2018-2022/2021-
2022.csv')
LonAm202122.info()
LonAm202122.sample(6, random_state=10)
```

```
             Date      Time  Easting  Northing        OAs
LSOA  \
54365  2021-12-08  12:25:03      NaN       NaN  E00023591  E01004677

2072   2021-01-11  23:18:31      NaN       NaN  E00023774  E01004710

58291  2022-01-01  02:47:47      NaN       NaN  E00023930  E01004735

12780  2021-03-26  04:36:23      NaN       NaN  E00175244  E01004701

33503  2021-07-31  18:02:56      NaN       NaN  E00023455  E01033607

23945  2021-06-06  06:03:47      NaN       NaN  E00023627  E01004686


              LSOA Name       Ward      Ward Name  MPS SNT  ...  \
54365  Westminster 005C  E05000635    Harrow Road   00BKGF  ...   
2072   Westminster 002D  E05000640     Maida Vale   00BKGL  ...   
58291  Westminster 018B  E05000644    St. James's  00BK07N  ...   
12780  Westminster 007A  E05000639  Little Venice  00BK11N  ...   
33503  Westminster 014G  E05000631      Bayswater   00BKGB  ...   
23945  Westminster 015E  E05000636      Hyde Park  00BK08N  ...   

                Incident Knife injury Overdose Self-harm injury
Sex  \
54365  Illness (unknown)          NaN      NaN              NaN
Female
2072   Illness (unknown)          NaN      NaN              NaN
```

```
Male
58291  Illness (unknown)          NaN      NaN                NaN
Unknown
12780  Illness (unknown)          NaN      NaN                NaN
Female
33503  Illness (unknown)          NaN      NaN                NaN
Male
23945  Illness (unknown)          NaN      NaN                NaN
Male

      Sex injury ~safestats theme1 ~safestats theme2  \
54365        NaN               NaN               NaN
2072         NaN               NaN               NaN
58291        NaN   Alcohol-Related               NaN
12780        NaN               NaN               NaN
33503        NaN               NaN               NaN
23945        NaN               NaN               NaN

      ~safestats crime category ~ SafeStats Crime SubGroup
54365                       NaN                        NaN
2072                        NaN                        NaN
58291                       NaN                        NaN
12780                       NaN                        NaN
33503                       NaN                        NaN
23945                       NaN                        NaN

[6 rows x 44 columns]
```

```
LonAM2 = LonAm202122.drop(columns=['Date','Time', 'Easting',
'Northing', 'LSOA', 'LSOA Name', 'Ward', 'Ward Name', 'MPS SNT', 'MPS
SNT Name', 'MPS NHD', 'MPS NHD Name', 'Borough', 'Borough Name',
'Police BCU (pre 2018)', 'Police BCU (pre 2018) Name', 'Police BCU
Sector', 'Police BCU Sector Name', 'Police BCU (post 2018)', 'Police
BCU (post 2018) Name', 'Age', 'Age group', 'Age group (youth)', 'Ampds
determinant', 'Vehicle id', 'Chief complaint', 'Illness', 'Alcohol
related', 'Assault injury', 'Class a related', 'Gun injury', 'Knife
injury', 'Overdose', 'Self-harm injury', 'Sex injury', 'Destination
hospital', 'Incidentid', 'Sex', '~safestats theme1', '~safestats
theme2', '~safestats crime category', '~ SafeStats Crime SubGroup'])
print(LonAM2)
LonAM2.sample(33, random_state=10)
```

```
             OAs           Incident
54365  E00023591  Illness (unknown)
2072   E00023774  Illness (unknown)
58291  E00023930  Illness (unknown)
12780  E00175244  Illness (unknown)
33503  E00023455  Illness (unknown)
23945  E00023627  Illness (unknown)
59907  E00024097  Illness (unknown)
```

```
64267  E00023937  Illness (unknown)
11454  E00023930    Police incident
6928   E00023931  Illness (unknown)
71984  E00023746  Illness (unknown)
25134  E00175249  Illness (unknown)
29892  E00023494    Illness (known)
58852  E00023776               Fall
66855  E00024130  Illness (unknown)
11789  E00024097    Illness (known)
64509  E00024133  Illness (unknown)
53978  E00023946               Fall
69304  E00024130    Illness (known)
15659  E00023892  Illness (unknown)
26750  E00023841  Illness (unknown)
474    E00023468    Illness (known)
41627  E00175256  Illness (unknown)
62651  E00023919    Illness (known)
72428  E00023489    Illness (known)
35367  E00023425  Illness (unknown)
77046  E00023929  Illness (unknown)
74220  E00024028  Illness (unknown)
774    E00023612  Illness (unknown)
31658  E00023802  Illness (unknown)
34006  E00023612  Illness (unknown)
44229  E00023480  Illness (unknown)
7178   E00023814          Self-harm
```

```
LonAM2 = LonAM2.dropna(axis=1,how='all')
```

```
LonAM2['Incident2122_no'] = 1
LonAM2.sample(33, random_state=10)
```

```
             OAs           Incident  Incident2122_no
54365  E00023591  Illness (unknown)                1
2072   E00023774  Illness (unknown)                1
58291  E00023930  Illness (unknown)                1
12780  E00175244  Illness (unknown)                1
33503  E00023455  Illness (unknown)                1
23945  E00023627  Illness (unknown)                1
59907  E00024097  Illness (unknown)                1
64267  E00023937  Illness (unknown)                1
11454  E00023930    Police incident                1
6928   E00023931  Illness (unknown)                1
71984  E00023746  Illness (unknown)                1
25134  E00175249  Illness (unknown)                1
29892  E00023494    Illness (known)                1
58852  E00023776               Fall                1
66855  E00024130  Illness (unknown)                1
11789  E00024097    Illness (known)                1
64509  E00024133  Illness (unknown)                1
53978  E00023946               Fall                1
```

```
69304  E00024130    Illness (known)                1
15659  E00023892  Illness (unknown)                1
26750  E00023841  Illness (unknown)                1
474    E00023468    Illness (known)                1
41627  E00175256  Illness (unknown)                1
62651  E00023919    Illness (known)                1
72428  E00023489    Illness (known)                1
35367  E00023425  Illness (unknown)                1
77046  E00023929  Illness (unknown)                1
74220  E00024028  Illness (unknown)                1
774    E00023612  Illness (unknown)                1
31658  E00023802  Illness (unknown)                1
34006  E00023612  Illness (unknown)                1
44229  E00023480  Illness (unknown)                1
7178   E00023814          Self-harm                1
```

```
LonAM1final = LonAM1.drop(columns=['Incident'])
LonAM2final = LonAM2.drop(columns=['Incident'])
```

```
LonAmALL = pd.merge(LonAM1final, LonAM2final, on=["OAs"])
print(LonAmALL)
```

```
                OAs  Incident1820_no  Incident2122_no
0         E00023937                1                1
1         E00023937                1                1
2         E00023937                1                1
3         E00023937                1                1
4         E00023937                1                1
...             ...              ...              ...
56711767  E00004772                1                1
56711768  E00004772                1                1
56711769  E00004772                1                1
56711770  E00004772                1                1
56711771  E00014118                1                1

[56711772 rows x 3 columns]
```

```
LonAmALL.sample(33, random_state=10)
LonAmALL.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 56711772 entries, 0 to 56711771
Data columns (total 3 columns):
 #   Column           Dtype
---  ------           -----
 0   OAs              object
 1   Incident1820_no  int64
 2   Incident2122_no  int64
dtypes: int64(2), object(1)
memory usage: 1.7+ GB
```

```
LonAmALL['LonAmALL']= LonAmALL['Incident1820_no'] +
LonAmALL['Incident2122_no']
print(LonAmALL)
```

```
                OAs  Incident1820_no  Incident2122_no  LonAmALL
0         E00023937                1                1         2
1         E00023937                1                1         2
2         E00023937                1                1         2
3         E00023937                1                1         2
4         E00023937                1                1         2
...             ...              ...              ...       ...
56711767  E00004772                1                1         2
56711768  E00004772                1                1         2
56711769  E00004772                1                1         2
56711770  E00004772                1                1         2
56711771  E00014118                1                1         2

[56711772 rows x 4 columns]
```

```
LonAmALL_trial=LonAmALL.groupby(by=['OAs'],
dropna=False).sum().reset_index()
```

```
LonAmALL_trial.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 794 entries, 0 to 793
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   OAs              794 non-null    object
 1   Incident1820_no  794 non-null    int64 
 2   Incident2122_no  794 non-null    int64 
 3   LonAmALL         794 non-null    int64 
dtypes: int64(3), object(1)
memory usage: 24.9+ KB
```

```
LonAmALL_trial.sample(16, random_state=10)
```

```
           OAs  Incident1820_no  Incident2122_no  LonAmALL
124  E00023528             1000             1000      2000
386  E00023810             1066             1066      2132
163  E00023570            33600            33600     67200
52   E00023455             3213             3213      6426
27   E00023430              551              551      1102
334  E00023758             2698             2698      5396
45   E00023448              994              994      1988
683  E00024127            51496            51496    102992
488  E00023914             1705             1705      3410
191  E00023602             3444             3444      6888
217  E00023629             1316             1316      2632
522  E00023950            26500            26500     53000
```

```
309  E00023729             3800             3800     7600
233  E00023647            10430            10430    20860
295  E00023715            19305            19305    38610
193  E00023604            40404            40404    80808
```

```
LonAmALL_trial =
LonAmALL_trial.drop(columns=['Incident1820_no','Incident2122_no'])
```

```
LonAmALL.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 56711772 entries, 0 to 56711771
Data columns (total 2 columns):
 #   Column    Dtype 
---  ------    ----- 
 0   OAs       object
 1   LonAmALL  int64 
dtypes: int64(1), object(1)
memory usage: 1.3+ GB
```

```
LonAmALL.to_csv('Ambulance.csv', encoding='utf-8', index=False)
```

Crime Data

```
Damage_new = Damage.drop(columns=['Dataset','~ SafeStats Crime
SubGroup', 'Date','Time', 'Easting', 'Northing', 'LSOA', 'LSOA Name',
'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name', 'MPS NHD', 'MPS NHD
Name', 'Borough', 'Borough Name', 'Police BCU (pre 2018)', 'Police BCU
(pre 2018) Name', 'Police BCU Sector', 'Police BCU Sector Name',
'Police BCU (post 2018)', 'Police BCU (post 2018) Name', 'LAS Incident
ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(Damage_new)
Damage_new.sample(33, random_state=10)
```

```
                 ~ SafeStats Crime Group        OAs
5183   Arson And Criminal Damage-Related  E00024137
835    Arson And Criminal Damage-Related  E00023931
12788  Arson And Criminal Damage-Related  E00023578
11085  Arson And Criminal Damage-Related  E00023947
5620   Arson And Criminal Damage-Related  E00024136
237    Arson And Criminal Damage-Related  E00023789
11673  Arson And Criminal Damage-Related  E00023774
13061  Arson And Criminal Damage-Related        NaN
13331  Arson And Criminal Damage-Related  E00175206
9041   Arson And Criminal Damage-Related  E00175219
11468  Arson And Criminal Damage-Related  E00023618
7585   Arson And Criminal Damage-Related  E00023686
5287   Arson And Criminal Damage-Related  E00023947
4884   Arson And Criminal Damage-Related  E00023727
11390  Arson And Criminal Damage-Related  E00024001
5817   Arson And Criminal Damage-Related  E00023928
4963   Arson And Criminal Damage-Related  E00023839
```

```
7020   Arson And Criminal Damage-Related  E00175191
5398   Arson And Criminal Damage-Related  E00175191
3956   Arson And Criminal Damage-Related  E00175241
12602  Arson And Criminal Damage-Related  E00023676
8466   Arson And Criminal Damage-Related  E00024103
13582  Arson And Criminal Damage-Related        NaN
2174   Arson And Criminal Damage-Related  E00023570
1810   Arson And Criminal Damage-Related  E00023438
8304   Arson And Criminal Damage-Related  E00024133
6047   Arson And Criminal Damage-Related  E00023418
145    Arson And Criminal Damage-Related  E00023519
4714   Arson And Criminal Damage-Related  E00023479
374    Arson And Criminal Damage-Related  E00023935
9022   Arson And Criminal Damage-Related  E00023483
7014   Arson And Criminal Damage-Related  E00024129
1029   Arson And Criminal Damage-Related  E00023490
```

```
Damage_new['Damage_incident'] = 1
Damage_new.sample(33, random_state=10)
```

```
                 ~ SafeStats Crime Group        OAs  Damage_incident
5183   Arson And Criminal Damage-Related  E00024137                1
835    Arson And Criminal Damage-Related  E00023931                1
12788  Arson And Criminal Damage-Related  E00023578                1
11085  Arson And Criminal Damage-Related  E00023947                1
5620   Arson And Criminal Damage-Related  E00024136                1
237    Arson And Criminal Damage-Related  E00023789                1
11673  Arson And Criminal Damage-Related  E00023774                1
13061  Arson And Criminal Damage-Related        NaN                1
13331  Arson And Criminal Damage-Related  E00175206                1
9041   Arson And Criminal Damage-Related  E00175219                1
11468  Arson And Criminal Damage-Related  E00023618                1
7585   Arson And Criminal Damage-Related  E00023686                1
5287   Arson And Criminal Damage-Related  E00023947                1
4884   Arson And Criminal Damage-Related  E00023727                1
11390  Arson And Criminal Damage-Related  E00024001                1
5817   Arson And Criminal Damage-Related  E00023928                1
4963   Arson And Criminal Damage-Related  E00023839                1
7020   Arson And Criminal Damage-Related  E00175191                1
5398   Arson And Criminal Damage-Related  E00175191                1
3956   Arson And Criminal Damage-Related  E00175241                1
12602  Arson And Criminal Damage-Related  E00023676                1
8466   Arson And Criminal Damage-Related  E00024103                1
13582  Arson And Criminal Damage-Related        NaN                1
2174   Arson And Criminal Damage-Related  E00023570                1
1810   Arson And Criminal Damage-Related  E00023438                1
8304   Arson And Criminal Damage-Related  E00024133                1
6047   Arson And Criminal Damage-Related  E00023418                1
145    Arson And Criminal Damage-Related  E00023519                1
4714   Arson And Criminal Damage-Related  E00023479                1
374    Arson And Criminal Damage-Related  E00023935                1
```

```
9022   Arson And Criminal Damage-Related  E00023483                1
7014   Arson And Criminal Damage-Related  E00024129                1
1029   Arson And Criminal Damage-Related  E00023490                1
```

```
Damage_new = Damage_new.drop(columns=['~ SafeStats Crime Group'])
Damage_new.sample(33, random_state=10)
```

```
             OAs  Damage_incident
5183   E00024137                1
835    E00023931                1
12788  E00023578                1
11085  E00023947                1
5620   E00024136                1
237    E00023789                1
11673  E00023774                1
13061        NaN                1
13331  E00175206                1
9041   E00175219                1
11468  E00023618                1
7585   E00023686                1
5287   E00023947                1
4884   E00023727                1
11390  E00024001                1
5817   E00023928                1
4963   E00023839                1
7020   E00175191                1
5398   E00175191                1
3956   E00175241                1
12602  E00023676                1
8466   E00024103                1
13582        NaN                1
2174   E00023570                1
1810   E00023438                1
8304   E00024133                1
6047   E00023418                1
145    E00023519                1
4714   E00023479                1
374    E00023935                1
9022   E00023483                1
7014   E00024129                1
1029   E00023490                1
```

```
Damage_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 14077 entries, 0 to 14076
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   OAs              13981 non-null  object
 1   Damage_incident  14077 non-null  int64
```

```
dtypes: int64(1), object(1)
memory usage: 220.1+ KB
```

```
Burglary_new = Burglary.drop(columns=['Dataset','~ SafeStats Crime
SubGroup', 'Date','Time', 'Easting', 'Northing', 'LSOA', 'LSOA Name',
'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name', 'MPS NHD', 'MPS NHD
Name', 'Borough', 'Borough Name', 'Police BCU (pre 2018)', 'Police BCU
(pre 2018) Name', 'Police BCU Sector', 'Police BCU Sector Name',
'Police BCU (post 2018)', 'Police BCU (post 2018) Name', 'LAS Incident
ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(Burglary_new)
Burglary_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs
9236         Burglary-Related  E00023777
12709        Burglary-Related  E00023945
14995        Burglary-Related  E00023824
11787        Burglary-Related  E00175237
6523         Burglary-Related  E00023595
11970        Burglary-Related  E00023935
15823        Burglary-Related  E00023579
13879        Burglary-Related  E00023953
8385         Burglary-Related  E00175266
19145        Burglary-Related  E00023698
18517        Burglary-Related  E00023543
6713         Burglary-Related  E00023575
5189         Burglary-Related  E00023937
6012         Burglary-Related  E00023544
11583        Burglary-Related  E00175225
12832        Burglary-Related  E00023691
10602        Burglary-Related  E00023934
2625         Burglary-Related  E00175194
5958         Burglary-Related  E00023928
8648         Burglary-Related  E00175190
8705         Burglary-Related  E00024113
16905        Burglary-Related  E00023744
17887        Burglary-Related  E00023432
16494        Burglary-Related  E00024091
12396        Burglary-Related  E00023623
12030        Burglary-Related  E00175261
8771         Burglary-Related  E00175259
10028        Burglary-Related  E00023415
5473         Burglary-Related  E00023893
967          Burglary-Related  E00175190
15952        Burglary-Related  E00024079
5587         Burglary-Related  E00023727
18080        Burglary-Related  E00024096
```

```
Burglary_new['Burglary_incident'] = 1
Burglary_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs  Burglary_incident
9236         Burglary-Related  E00023777                  1
12709        Burglary-Related  E00023945                  1
14995        Burglary-Related  E00023824                  1
11787        Burglary-Related  E00175237                  1
6523         Burglary-Related  E00023595                  1
11970        Burglary-Related  E00023935                  1
15823        Burglary-Related  E00023579                  1
13879        Burglary-Related  E00023953                  1
8385         Burglary-Related  E00175266                  1
19145        Burglary-Related  E00023698                  1
18517        Burglary-Related  E00023543                  1
6713         Burglary-Related  E00023575                  1
5189         Burglary-Related  E00023937                  1
6012         Burglary-Related  E00023544                  1
11583        Burglary-Related  E00175225                  1
12832        Burglary-Related  E00023691                  1
10602        Burglary-Related  E00023934                  1
2625         Burglary-Related  E00175194                  1
5958         Burglary-Related  E00023928                  1
8648         Burglary-Related  E00175190                  1
8705         Burglary-Related  E00024113                  1
16905        Burglary-Related  E00023744                  1
17887        Burglary-Related  E00023432                  1
16494        Burglary-Related  E00024091                  1
12396        Burglary-Related  E00023623                  1
12030        Burglary-Related  E00175261                  1
8771         Burglary-Related  E00175259                  1
10028        Burglary-Related  E00023415                  1
5473         Burglary-Related  E00023893                  1
967          Burglary-Related  E00175190                  1
15952        Burglary-Related  E00024079                  1
5587         Burglary-Related  E00023727                  1
18080        Burglary-Related  E00024096                  1
```

```
Burglary_new = Burglary_new.drop(columns=['~ SafeStats Crime Group'])
Burglary_new.sample(33, random_state=10)
```

```
             OAs  Burglary_incident
9236   E00023777                  1
12709  E00023945                  1
14995  E00023824                  1
11787  E00175237                  1
6523   E00023595                  1
11970  E00023935                  1
15823  E00023579                  1
13879  E00023953                  1
8385   E00175266                  1
19145  E00023698                  1
18517  E00023543                  1
6713   E00023575                  1
```

```
5189   E00023937                  1
6012   E00023544                  1
11583  E00175225                  1
12832  E00023691                  1
10602  E00023934                  1
2625   E00175194                  1
5958   E00023928                  1
8648   E00175190                  1
8705   E00024113                  1
16905  E00023744                  1
17887  E00023432                  1
16494  E00024091                  1
12396  E00023623                  1
12030  E00175261                  1
8771   E00175259                  1
10028  E00023415                  1
5473   E00023893                  1
967    E00175190                  1
15952  E00024079                  1
5587   E00023727                  1
18080  E00024096                  1
```

```
Burglary_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 19994 entries, 0 to 19993
Data columns (total 2 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   OAs                19940 non-null  object
 1   Burglary_incident  19994 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 312.5+ KB
```

```
Disorder_new = Disorder.drop(columns=['Dataset','~ SafeStats Crime 
SubGroup', 'Date','Time', 'Easting', 'Northing', 'LSOA', 'LSOA Name', 
'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name', 'MPS NHD', 'MPS NHD 
Name', 'Borough', 'Borough Name', 'Police BCU (pre 2018)', 'Police BCU 
(pre 2018) Name', 'Police BCU Sector', 'Police BCU Sector Name', 
'Police BCU (post 2018)', 'Police BCU (post 2018) Name', 'LAS Incident 
ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(Disorder_new)
Disorder_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs
75414        Disorder-Related  E00023945
56552        Disorder-Related  E00023570
36674        Disorder-Related  E00024133
54061        Disorder-Related  E00023647
45249        Disorder-Related  E00023724
18681        Disorder-Related  E00023845
```

```
64947        Disorder-Related  E00024129
8498         Disorder-Related  E00023941
33306        Disorder-Related  E00023818
5041         Disorder-Related  E00023618
70075        Disorder-Related  E00175233
27919        Disorder-Related  E00023689
65595        Disorder-Related  E00175249
23372        Disorder-Related  E00175190
24832        Disorder-Related  E00023698
53616        Disorder-Related  E00023787
41031        Disorder-Related  E00023934
72090        Disorder-Related  E00023543
66081        Disorder-Related  E00023935
24137        Disorder-Related  E00023893
74782        Disorder-Related  E00024079
57096        Disorder-Related  E00024030
71292        Disorder-Related  E00023586
21010        Disorder-Related  E00175249
20109        Disorder-Related  E00023768
50377        Disorder-Related  E00024113
45571        Disorder-Related  E00023945
66819        Disorder-Related  E00023930
23198        Disorder-Related  E00023506
41218        Disorder-Related  E00023935
3960         Disorder-Related  E00024080
25182        Disorder-Related  E00175180
7539         Disorder-Related  E00023566
```

```
Disorder_new['Disorder_incident'] = 1
Disorder_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs  Disorder_incident
75414        Disorder-Related  E00023945                  1
56552        Disorder-Related  E00023570                  1
36674        Disorder-Related  E00024133                  1
54061        Disorder-Related  E00023647                  1
45249        Disorder-Related  E00023724                  1
18681        Disorder-Related  E00023845                  1
64947        Disorder-Related  E00024129                  1
8498         Disorder-Related  E00023941                  1
33306        Disorder-Related  E00023818                  1
5041         Disorder-Related  E00023618                  1
70075        Disorder-Related  E00175233                  1
27919        Disorder-Related  E00023689                  1
65595        Disorder-Related  E00175249                  1
23372        Disorder-Related  E00175190                  1
24832        Disorder-Related  E00023698                  1
53616        Disorder-Related  E00023787                  1
41031        Disorder-Related  E00023934                  1
72090        Disorder-Related  E00023543                  1
66081        Disorder-Related  E00023935                  1
```

```
24137        Disorder-Related  E00023893                  1
74782        Disorder-Related  E00024079                  1
57096        Disorder-Related  E00024030                  1
71292        Disorder-Related  E00023586                  1
21010        Disorder-Related  E00175249                  1
20109        Disorder-Related  E00023768                  1
50377        Disorder-Related  E00024113                  1
45571        Disorder-Related  E00023945                  1
66819        Disorder-Related  E00023930                  1
23198        Disorder-Related  E00023506                  1
41218        Disorder-Related  E00023935                  1
3960         Disorder-Related  E00024080                  1
25182        Disorder-Related  E00175180                  1
7539         Disorder-Related  E00023566                  1
```

```
Disorder_new = Disorder_new.drop(columns=['~ SafeStats Crime Group'])
Disorder_new.sample(33, random_state=10)
```

```
             OAs  Disorder_incident
75414  E00023945                  1
56552  E00023570                  1
36674  E00024133                  1
54061  E00023647                  1
45249  E00023724                  1
18681  E00023845                  1
64947  E00024129                  1
8498   E00023941                  1
33306  E00023818                  1
5041   E00023618                  1
70075  E00175233                  1
27919  E00023689                  1
65595  E00175249                  1
23372  E00175190                  1
24832  E00023698                  1
53616  E00023787                  1
41031  E00023934                  1
72090  E00023543                  1
66081  E00023935                  1
24137  E00023893                  1
74782  E00024079                  1
57096  E00024030                  1
71292  E00023586                  1
21010  E00175249                  1
20109  E00023768                  1
50377  E00024113                  1
45571  E00023945                  1
66819  E00023930                  1
23198  E00023506                  1
41218  E00023935                  1
3960   E00024080                  1
```

```
25182  E00175180                  1
7539   E00023566                  1
```

```
Disorder_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 78977 entries, 0 to 78976
Data columns (total 2 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   OAs                78584 non-null  object
 1   Disorder_incident  78977 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 1.2+ MB
```

```
Fraud_new = Fraud.drop(columns=['Dataset','~ SafeStats Crime
SubGroup', 'Date','Time', 'Easting', 'Northing', 'LSOA', 'LSOA Name',
'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name', 'MPS NHD', 'MPS NHD
Name', 'Borough', 'Borough Name', 'Police BCU (pre 2018)', 'Police BCU
(pre 2018) Name', 'Police BCU Sector', 'Police BCU Sector Name',
'Police BCU (post 2018)', 'Police BCU (post 2018) Name', 'LAS Incident
ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(Fraud_new)
Fraud_new.sample(33, random_state=10)
```

```
     ~ SafeStats Crime Group        OAs
94             Fraud-Related  E00024048
1841           Fraud-Related  E00023929
211            Fraud-Related  E00175186
952            Fraud-Related  E00023936
1679           Fraud-Related  E00175214
617            Fraud-Related  E00023839
470            Fraud-Related  E00023818
1743           Fraud-Related  E00023517
1084           Fraud-Related  E00023649
339            Fraud-Related  E00024048
990            Fraud-Related  E00023936
994            Fraud-Related  E00175237
1758           Fraud-Related  E00175190
589            Fraud-Related  E00023935
1874           Fraud-Related  E00024114
1910           Fraud-Related  E00023681
1152           Fraud-Related  E00175190
609            Fraud-Related  E00023643
1161           Fraud-Related  E00175214
1646           Fraud-Related  E00023935
36             Fraud-Related  E00024048
904            Fraud-Related  E00175214
805            Fraud-Related  E00023935
1820           Fraud-Related  E00175191
1849           Fraud-Related  E00024092
```

```
96              Fraud-Related  E00024048
163             Fraud-Related  E00175190
1401            Fraud-Related  E00175190
895             Fraud-Related  E00023935
1559            Fraud-Related  E00023699
426             Fraud-Related  E00175190
27              Fraud-Related  E00024048
815             Fraud-Related  E00024114
```

```
Fraud_new['Fraud_incident'] = 1
Fraud_new.sample(33, random_state=10)
```

```
     ~ SafeStats Crime Group        OAs  Fraud_incident
94              Fraud-Related  E00024048               1
1841            Fraud-Related  E00023929               1
211             Fraud-Related  E00175186               1
952             Fraud-Related  E00023936               1
1679            Fraud-Related  E00175214               1
617             Fraud-Related  E00023839               1
470             Fraud-Related  E00023818               1
1743            Fraud-Related  E00023517               1
1084            Fraud-Related  E00023649               1
339             Fraud-Related  E00024048               1
990             Fraud-Related  E00023936               1
994             Fraud-Related  E00175237               1
1758            Fraud-Related  E00175190               1
589             Fraud-Related  E00023935               1
1874            Fraud-Related  E00024114               1
1910            Fraud-Related  E00023681               1
1152            Fraud-Related  E00175190               1
609             Fraud-Related  E00023643               1
1161            Fraud-Related  E00175214               1
1646            Fraud-Related  E00023935               1
36              Fraud-Related  E00024048               1
904             Fraud-Related  E00175214               1
805             Fraud-Related  E00023935               1
1820            Fraud-Related  E00175191               1
1849            Fraud-Related  E00024092               1
96              Fraud-Related  E00024048               1
163             Fraud-Related  E00175190               1
1401            Fraud-Related  E00175190               1
895             Fraud-Related  E00023935               1
1559            Fraud-Related  E00023699               1
426             Fraud-Related  E00175190               1
27              Fraud-Related  E00024048               1
815             Fraud-Related  E00024114               1
```

```
Fraud_new = Fraud_new.drop(columns=['~ SafeStats Crime Group'])
Fraud_new.sample(33, random_state=10)
```

```
            OAs  Fraud_incident
94    E00024048               1
1841  E00023929               1
211   E00175186               1
952   E00023936               1
1679  E00175214               1
617   E00023839               1
470   E00023818               1
1743  E00023517               1
1084  E00023649               1
339   E00024048               1
990   E00023936               1
994   E00175237               1
1758  E00175190               1
589   E00023935               1
1874  E00024114               1
1910  E00023681               1
1152  E00175190               1
609   E00023643               1
1161  E00175214               1
1646  E00023935               1
36    E00024048               1
904   E00175214               1
805   E00023935               1
1820  E00175191               1
1849  E00024092               1
96    E00024048               1
163   E00175190               1
1401  E00175190               1
895   E00023935               1
1559  E00023699               1
426   E00175190               1
27    E00024048               1
815   E00024114               1
```

```
Fraud_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 1960 entries, 0 to 1959
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   OAs             1952 non-null   object
 1   Fraud_incident  1960 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 30.8+ KB
```

```
Robbery_new = Robbery.drop(columns=['Dataset','~ SafeStats Crime
SubGroup', 'Date','Time', 'Easting', 'Northing', 'LSOA', 'LSOA Name',
'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name', 'MPS NHD', 'MPS NHD
Name', 'Borough', 'Borough Name', 'Police BCU (pre 2018)', 'Police BCU
(pre 2018) Name', 'Police BCU Sector', 'Police BCU Sector Name',
'Police BCU (post 2018)', 'Police BCU (post 2018) Name', 'LAS Incident
ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(Robbery_new)
Robbery_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs
5711          Robbery-Related  E00023801
9712          Robbery-Related  E00024133
3841          Robbery-Related  E00175206
14612         Robbery-Related  E00024138
4291          Robbery-Related  E00023937
16573         Robbery-Related  E00023929
13291         Robbery-Related  E00023948
10314         Robbery-Related  E00024107
13108         Robbery-Related  E00023617
8413          Robbery-Related  E00024138
99            Robbery-Related  E00024048
16506         Robbery-Related  E00023648
10783         Robbery-Related  E00023496
16575         Robbery-Related  E00024032
11628         Robbery-Related  E00023864
15613         Robbery-Related  E00024105
16827         Robbery-Related  E00024130
4090          Robbery-Related  E00023648
13334         Robbery-Related  E00024134
8654          Robbery-Related  E00024114
3475          Robbery-Related  E00023960
12947         Robbery-Related  E00023928
11668         Robbery-Related  E00023963
2189          Robbery-Related  E00023522
825           Robbery-Related  E00175192
605           Robbery-Related  E00024073
3556          Robbery-Related  E00023928
3675          Robbery-Related  E00023596
12735         Robbery-Related  E00175194
12099         Robbery-Related  E00023702
12297         Robbery-Related  E00024129
16088         Robbery-Related  E00023627
15361         Robbery-Related  E00175214
```

```
Robbery_new['Robbery_incident'] = 1
Robbery_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs  Robbery_incident
5711          Robbery-Related  E00023801                 1
9712          Robbery-Related  E00024133                 1
3841          Robbery-Related  E00175206                 1
14612         Robbery-Related  E00024138                 1
4291          Robbery-Related  E00023937                 1
16573         Robbery-Related  E00023929                 1
```

```
13291         Robbery-Related  E00023948                 1
10314         Robbery-Related  E00024107                 1
13108         Robbery-Related  E00023617                 1
8413          Robbery-Related  E00024138                 1
99            Robbery-Related  E00024048                 1
16506         Robbery-Related  E00023648                 1
10783         Robbery-Related  E00023496                 1
16575         Robbery-Related  E00024032                 1
11628         Robbery-Related  E00023864                 1
15613         Robbery-Related  E00024105                 1
16827         Robbery-Related  E00024130                 1
4090          Robbery-Related  E00023648                 1
13334         Robbery-Related  E00024134                 1
8654          Robbery-Related  E00024114                 1
3475          Robbery-Related  E00023960                 1
12947         Robbery-Related  E00023928                 1
11668         Robbery-Related  E00023963                 1
2189          Robbery-Related  E00023522                 1
825           Robbery-Related  E00175192                 1
605           Robbery-Related  E00024073                 1
3556          Robbery-Related  E00023928                 1
3675          Robbery-Related  E00023596                 1
12735         Robbery-Related  E00175194                 1
12099         Robbery-Related  E00023702                 1
12297         Robbery-Related  E00024129                 1
16088         Robbery-Related  E00023627                 1
15361         Robbery-Related  E00175214                 1
```

```
Robbery_new = Robbery_new.drop(columns=['~ SafeStats Crime Group'])
Robbery_new.sample(33, random_state=10)
```

```
             OAs  Robbery_incident
5711   E00023801                 1
9712   E00024133                 1
3841   E00175206                 1
14612  E00024138                 1
4291   E00023937                 1
16573  E00023929                 1
13291  E00023948                 1
10314  E00024107                 1
13108  E00023617                 1
8413   E00024138                 1
99     E00024048                 1
16506  E00023648                 1
10783  E00023496                 1
16575  E00024032                 1
11628  E00023864                 1
15613  E00024105                 1
16827  E00024130                 1
4090   E00023648                 1
13334  E00024134                 1
```

```
8654   E00024114                 1
3475   E00023960                 1
12947  E00023928                 1
11668  E00023963                 1
2189   E00023522                 1
825    E00175192                 1
605    E00024073                 1
3556   E00023928                 1
3675   E00023596                 1
12735  E00175194                 1
12099  E00023702                 1
12297  E00024129                 1
16088  E00023627                 1
15361  E00175214                 1
```

```
Robbery_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 16828 entries, 0 to 16827
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   OAs               16670 non-null  object
 1   Robbery_incident  16828 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 263.1+ KB
```

```
SexRelated_new = SexRelated.drop(columns=['Dataset','~ SafeStats Crime 
SubGroup', 'Date','Time', 'Easting', 'Northing', 'LSOA', 'LSOA Name', 
'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name', 'MPS NHD', 'MPS NHD 
Name', 'Borough', 'Borough Name', 'Police BCU (pre 2018)', 'Police BCU 
(pre 2018) Name', 'Police BCU Sector', 'Police BCU Sector Name', 
'Police BCU (post 2018)', 'Police BCU (post 2018) Name', 'LAS Incident 
ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(SexRelated_new)
SexRelated_new.sample(33, random_state=10)
```

```
     ~ SafeStats Crime Group        OAs
2384             Sex-Related        NaN
176              Sex-Related  E00023920
1525             Sex-Related        NaN
3011             Sex-Related        NaN
499              Sex-Related  E00024048
2137             Sex-Related        NaN
1234             Sex-Related        NaN
946              Sex-Related        NaN
1060             Sex-Related        NaN
2665             Sex-Related        NaN
1049             Sex-Related        NaN
692              Sex-Related  E00175186
966              Sex-Related        NaN
```

```
145              Sex-Related  E00024113
1581             Sex-Related        NaN
348              Sex-Related  E00175190
1247             Sex-Related        NaN
3274             Sex-Related        NaN
1276             Sex-Related        NaN
1670             Sex-Related        NaN
1676             Sex-Related        NaN
1018             Sex-Related        NaN
1825             Sex-Related        NaN
2366             Sex-Related        NaN
1145             Sex-Related        NaN
2440             Sex-Related        NaN
556              Sex-Related  E00023936
3180             Sex-Related        NaN
2747             Sex-Related        NaN
1357             Sex-Related        NaN
893              Sex-Related        NaN
1894             Sex-Related        NaN
2266             Sex-Related        NaN
```

```
SexRelated_new['SexRelated_incident'] = 1
SexRelated_new.sample(33, random_state=10)
```

```
     ~ SafeStats Crime Group        OAs  SexRelated_incident
2384             Sex-Related        NaN                    1
176              Sex-Related  E00023920                    1
1525             Sex-Related        NaN                    1
3011             Sex-Related        NaN                    1
499              Sex-Related  E00024048                    1
2137             Sex-Related        NaN                    1
1234             Sex-Related        NaN                    1
946              Sex-Related        NaN                    1
1060             Sex-Related        NaN                    1
2665             Sex-Related        NaN                    1
1049             Sex-Related        NaN                    1
692              Sex-Related  E00175186                    1
966              Sex-Related        NaN                    1
145              Sex-Related  E00024113                    1
1581             Sex-Related        NaN                    1
348              Sex-Related  E00175190                    1
1247             Sex-Related        NaN                    1
3274             Sex-Related        NaN                    1
1276             Sex-Related        NaN                    1
1670             Sex-Related        NaN                    1
1676             Sex-Related        NaN                    1
1018             Sex-Related        NaN                    1
1825             Sex-Related        NaN                    1
2366             Sex-Related        NaN                    1
1145             Sex-Related        NaN                    1
2440             Sex-Related        NaN                    1
```

```
556              Sex-Related  E00023936                1
3180             Sex-Related        NaN                1
2747             Sex-Related        NaN                1
1357             Sex-Related        NaN                1
893              Sex-Related        NaN                1
1894             Sex-Related        NaN                1
2266             Sex-Related        NaN                1
```

```
SexRelated_new = SexRelated_new.drop(columns=['~ SafeStats Crime
Group'])
SexRelated_new.sample(33, random_state=10)
```

```
            OAs  SexRelated_incident
2384        NaN                    1
176   E00023920                    1
1525        NaN                    1
3011        NaN                    1
499   E00024048                    1
2137        NaN                    1
1234        NaN                    1
946         NaN                    1
1060        NaN                    1
2665        NaN                    1
1049        NaN                    1
692   E00175186                    1
966         NaN                    1
145   E00024113                    1
1581        NaN                    1
348   E00175190                    1
1247        NaN                    1
3274        NaN                    1
1276        NaN                    1
1670        NaN                    1
1676        NaN                    1
1018        NaN                    1
1825        NaN                    1
2366        NaN                    1
1145        NaN                    1
2440        NaN                    1
556   E00023936                    1
3180        NaN                    1
2747        NaN                    1
1357        NaN                    1
893         NaN                    1
1894        NaN                    1
2266        NaN                    1
```

```
SexRelated_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 3548 entries, 0 to 3547
```

```
Data columns (total 2 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   OAs                   843 non-null    object
 1   SexRelated_incident   3548 non-null   int64
dtypes: int64(1), object(1)
memory usage: 55.6+ KB
```

```
Violence_new = Violence.drop(columns=['Dataset','~ SafeStats Crime
SubGroup', 'Date','Time', 'Easting', 'Northing', 'LSOA', 'LSOA Name',
'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name', 'MPS NHD', 'MPS NHD
Name', 'Borough', 'Borough Name', 'Police BCU (pre 2018)', 'Police BCU
(pre 2018) Name', 'Police BCU Sector', 'Police BCU Sector Name',
'Police BCU (post 2018)', 'Police BCU (post 2018) Name', 'LAS Incident
ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(Violence_new)
Violence_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs
47945        Violence-Related  E00024112
36215        Violence-Related  E00175190
61188        Violence-Related  E00023763
61201        Violence-Related  E00024028
43802        Violence-Related  E00024002
32140        Violence-Related  E00024130
36095        Violence-Related  E00023572
54607        Violence-Related  E00023960
64086        Violence-Related  E00175225
47196        Violence-Related  E00175269
63168        Violence-Related  E00175186
52371        Violence-Related  E00023580
21326        Violence-Related  E00023596
66046        Violence-Related        NaN
61817        Violence-Related        NaN
2187         Violence-Related  E00023928
2656         Violence-Related  E00024113
50563        Violence-Related  E00023415
59455        Violence-Related  E00023579
19758        Violence-Related  E00023565
59539        Violence-Related  E00023539
6317         Violence-Related  E00175187
8846         Violence-Related  E00023937
23380        Violence-Related  E00023945
2277         Violence-Related  E00024048
32400        Violence-Related  E00023415
5886         Violence-Related  E00024130
9401         Violence-Related  E00024134
23447        Violence-Related  E00023561
48176        Violence-Related  E00023936
67613        Violence-Related  E00175193
```

```
36417        Violence-Related  E00023935
13207        Violence-Related  E00023930
```

```
Violence_new['Violence_incident'] = 1
Violence_new.sample(33, random_state=10)
```

```
      ~ SafeStats Crime Group        OAs  Violence_incident
47945        Violence-Related  E00024112                  1
36215        Violence-Related  E00175190                  1
61188        Violence-Related  E00023763                  1
61201        Violence-Related  E00024028                  1
43802        Violence-Related  E00024002                  1
32140        Violence-Related  E00024130                  1
36095        Violence-Related  E00023572                  1
54607        Violence-Related  E00023960                  1
64086        Violence-Related  E00175225                  1
47196        Violence-Related  E00175269                  1
63168        Violence-Related  E00175186                  1
52371        Violence-Related  E00023580                  1
21326        Violence-Related  E00023596                  1
66046        Violence-Related        NaN                  1
61817        Violence-Related        NaN                  1
2187         Violence-Related  E00023928                  1
2656         Violence-Related  E00024113                  1
50563        Violence-Related  E00023415                  1
59455        Violence-Related  E00023579                  1
19758        Violence-Related  E00023565                  1
59539        Violence-Related  E00023539                  1
6317         Violence-Related  E00175187                  1
8846         Violence-Related  E00023937                  1
23380        Violence-Related  E00023945                  1
2277         Violence-Related  E00024048                  1
32400        Violence-Related  E00023415                  1
5886         Violence-Related  E00024130                  1
9401         Violence-Related  E00024134                  1
23447        Violence-Related  E00023561                  1
48176        Violence-Related  E00023936                  1
67613        Violence-Related  E00175193                  1
36417        Violence-Related  E00023935                  1
13207        Violence-Related  E00023930                  1
```

```
Violence_new = Violence_new.drop(columns=['~ SafeStats Crime Group'])
Violence_new.sample(33, random_state=10)
```

```
             OAs  Violence_incident
47945  E00024112                  1
36215  E00175190                  1
61188  E00023763                  1
61201  E00024028                  1
43802  E00024002                  1
32140  E00024130                  1
```

```
36095  E00023572                 1
54607  E00023960                 1
64086  E00175225                 1
47196  E00175269                 1
63168  E00175186                 1
52371  E00023580                 1
21326  E00023596                 1
66046        NaN                 1
61817        NaN                 1
2187   E00023928                 1
2656   E00024113                 1
50563  E00023415                 1
59455  E00023579                 1
19758  E00023565                 1
59539  E00023539                 1
6317   E00175187                 1
8846   E00023937                 1
23380  E00023945                 1
2277   E00024048                 1
32400  E00023415                 1
5886   E00024130                 1
9401   E00024134                 1
23447  E00023561                 1
48176  E00023936                 1
67613  E00175193                 1
36417  E00023935                 1
13207  E00023930                 1
```

```
Violence_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 71614 entries, 0 to 71613
Data columns (total 2 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   OAs                70430 non-null  object
 1   Violence_incident  71614 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 1.1+ MB
```

```
WeaponPossession_new = WeaponPossession.drop(columns=['Dataset','~
SafeStats Crime SubGroup', 'Date','Time', 'Easting', 'Northing',
'LSOA', 'LSOA Name', 'Ward', 'Ward Name', 'MPS SNT', 'MPS SNT Name',
'MPS NHD', 'MPS NHD Name', 'Borough', 'Borough Name', 'Police BCU (pre
2018)', 'Police BCU (pre 2018) Name', 'Police BCU Sector', 'Police BCU
Sector Name', 'Police BCU (post 2018)', 'Police BCU (post 2018) Name',
'LAS Incident ID', 'ED Incident ID', 'ED Incident Confidence Score'])
print(WeaponPossession_new)
WeaponPossession_new.sample(33, random_state=10)
```

```
        ~ SafeStats Crime Group        OAs
299   Weapon Possession-Related  E00023964
1196  Weapon Possession-Related  E00024114
1139  Weapon Possession-Related  E00023945
1869  Weapon Possession-Related  E00175190
1000  Weapon Possession-Related  E00024054
1972  Weapon Possession-Related  E00023935
1062  Weapon Possession-Related  E00023929
761   Weapon Possession-Related  E00023945
524   Weapon Possession-Related  E00023801
696   Weapon Possession-Related  E00023845
152   Weapon Possession-Related  E00023945
163   Weapon Possession-Related  E00023905
176   Weapon Possession-Related  E00175206
637   Weapon Possession-Related  E00175191
1040  Weapon Possession-Related  E00023860
1922  Weapon Possession-Related  E00024095
282   Weapon Possession-Related  E00023955
813   Weapon Possession-Related  E00024128
1626  Weapon Possession-Related  E00023935
1981  Weapon Possession-Related  E00023945
915   Weapon Possession-Related  E00023945
1149  Weapon Possession-Related  E00175275
924   Weapon Possession-Related  E00024058
1179  Weapon Possession-Related  E00024124
1665  Weapon Possession-Related  E00023670
1160  Weapon Possession-Related  E00023731
452   Weapon Possession-Related  E00023572
2038  Weapon Possession-Related  E00023635
2101  Weapon Possession-Related  E00023945
174   Weapon Possession-Related  E00023938
110   Weapon Possession-Related  E00023945
920   Weapon Possession-Related  E00023945
845   Weapon Possession-Related  E00175194
```

```
WeaponPossession_new['WeaponPossession_incident'] = 1
WeaponPossession_new.sample(33, random_state=10)
```

```
        ~ SafeStats Crime Group        OAs  WeaponPossession_incident
299   Weapon Possession-Related  E00023964                          1
1196  Weapon Possession-Related  E00024114                          1
1139  Weapon Possession-Related  E00023945                          1
1869  Weapon Possession-Related  E00175190                          1
1000  Weapon Possession-Related  E00024054                          1
1972  Weapon Possession-Related  E00023935                          1
1062  Weapon Possession-Related  E00023929                          1
761   Weapon Possession-Related  E00023945                          1
524   Weapon Possession-Related  E00023801                          1
696   Weapon Possession-Related  E00023845                          1
152   Weapon Possession-Related  E00023945                          1
163   Weapon Possession-Related  E00023905                          1
```

```
176   Weapon Possession-Related  E00175206                           1
637   Weapon Possession-Related  E00175191                           1
1040  Weapon Possession-Related  E00023860                           1
1922  Weapon Possession-Related  E00024095                           1
282   Weapon Possession-Related  E00023955                           1
813   Weapon Possession-Related  E00024128                           1
1626  Weapon Possession-Related  E00023935                           1
1981  Weapon Possession-Related  E00023945                           1
915   Weapon Possession-Related  E00023945                           1
1149  Weapon Possession-Related  E00175275                           1
924   Weapon Possession-Related  E00024058                           1
1179  Weapon Possession-Related  E00024124                           1
1665  Weapon Possession-Related  E00023670                           1
1160  Weapon Possession-Related  E00023731                           1
452   Weapon Possession-Related  E00023572                           1
2038  Weapon Possession-Related  E00023635                           1
2101  Weapon Possession-Related  E00023945                           1
174   Weapon Possession-Related  E00023938                           1
110   Weapon Possession-Related  E00023945                           1
920   Weapon Possession-Related  E00023945                           1
845   Weapon Possession-Related  E00175194                           1
```

```
WeaponPossession_new = WeaponPossession_new.drop(columns=['~ SafeStats Crime Group'])
WeaponPossession_new.sample(33, random_state=10)
```

```
            OAs  WeaponPossession_incident
299   E00023964                          1
1196  E00024114                          1
1139  E00023945                          1
1869  E00175190                          1
1000  E00024054                          1
1972  E00023935                          1
1062  E00023929                          1
761   E00023945                          1
524   E00023801                          1
696   E00023845                          1
152   E00023945                          1
163   E00023905                          1
176   E00175206                          1
637   E00175191                          1
1040  E00023860                          1
1922  E00024095                          1
282   E00023955                          1
813   E00024128                          1
1626  E00023935                          1
1981  E00023945                          1
915   E00023945                          1
1149  E00175275                          1
924   E00024058                          1
1179  E00024124                          1
```

```
1665  E00023670                          1
1160  E00023731                          1
452   E00023572                          1
2038  E00023635                          1
2101  E00023945                          1
174   E00023938                          1
110   E00023945                          1
920   E00023945                          1
845   E00175194                          1
```

```
WeaponPossession_new.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 2198 entries, 0 to 2197
Data columns (total 2 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   OAs                         2172 non-null   object
 1   WeaponPossession_incident  2198 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 34.5+ KB
```

```
Crime1 = Damage_new.append(Burglary_new)
Crime1.sample(15, random_state=10)
Crime1.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 34071 entries, 0 to 19993
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   OAs                 33921 non-null  object 
 1   Damage_incident     14077 non-null  float64
 2   Burglary_incident  19994 non-null  float64
dtypes: float64(2), object(1)
memory usage: 1.0+ MB
```

```
Crime2 = Crime1.append(Disorder_new)
Crime2.sample(15, random_state=10)
Crime2.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 113048 entries, 0 to 78976
Data columns (total 4 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   OAs                 112505 non-null  object 
 1   Damage_incident     14077 non-null   float64
 2   Burglary_incident  19994 non-null   float64
 3   Disorder_incident  78977 non-null   float64
```

```
dtypes: float64(3), object(1)
memory usage: 4.3+ MB
```

```
Crime3 = Crime2.append(Fraud_new)
Crime3.sample(15, random_state=10)
Crime3.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 115008 entries, 0 to 1959
Data columns (total 5 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   OAs                114457 non-null  object
 1   Damage_incident    14077 non-null   float64
 2   Burglary_incident  19994 non-null   float64
 3   Disorder_incident  78977 non-null   float64
 4   Fraud_incident     1960 non-null    float64
dtypes: float64(4), object(1)
memory usage: 5.3+ MB
```

```
Crime4 = Crime3.append(Robbery_new)
Crime4.sample(15, random_state=10)
Crime4.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 131836 entries, 0 to 16827
Data columns (total 6 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   OAs                131127 non-null  object
 1   Damage_incident    14077 non-null   float64
 2   Burglary_incident  19994 non-null   float64
 3   Disorder_incident  78977 non-null   float64
 4   Fraud_incident     1960 non-null    float64
 5   Robbery_incident   16828 non-null   float64
dtypes: float64(5), object(1)
memory usage: 7.0+ MB
```

```
Crime5 = Crime4.append(SexRelated_new)
Crime5.sample(15, random_state=10)
Crime5.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 135384 entries, 0 to 3547
Data columns (total 7 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   OAs                  131970 non-null  object
 1   Damage_incident      14077 non-null   float64
 2   Burglary_incident    19994 non-null   float64
 3   Disorder_incident    78977 non-null   float64
 4   Fraud_incident       1960 non-null    float64
 5   Robbery_incident     16828 non-null   float64
 6   SexRelated_incident  3548 non-null    float64
dtypes: float64(6), object(1)
memory usage: 8.3+ MB
```

```
Crime6 = Crime5.append(Violence_new)
Crime6.sample(15, random_state=10)
Crime6.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 206998 entries, 0 to 71613
Data columns (total 8 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   OAs                  202400 non-null  object 
 1   Damage_incident      14077 non-null   float64
 2   Burglary_incident    19994 non-null   float64
 3   Disorder_incident    78977 non-null   float64
 4   Fraud_incident       1960 non-null    float64
 5   Robbery_incident     16828 non-null   float64
 6   SexRelated_incident  3548 non-null    float64
 7   Violence_incident    71614 non-null   float64
dtypes: float64(7), object(1)
memory usage: 14.2+ MB
```

```
Crime_all = Crime6.append(WeaponPossession_new)
Crime_all.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 209196 entries, 0 to 2197
Data columns (total 9 columns):
 #   Column                     Non-Null Count   Dtype  
---  ------                     --------------   -----  
 0   OAs                        204572 non-null  object 
 1   Damage_incident            14077 non-null   float64
 2   Burglary_incident          19994 non-null   float64
 3   Disorder_incident          78977 non-null   float64
 4   Fraud_incident             1960 non-null    float64
 5   Robbery_incident           16828 non-null   float64
 6   SexRelated_incident        3548 non-null    float64
 7   Violence_incident          71614 non-null   float64
 8   WeaponPossession_incident  2198 non-null    float64
dtypes: float64(8), object(1)
memory usage: 16.0+ MB
```

```
Crime_all.sample(45, random_state=10)
```

```
             OAs  Damage_incident  Burglary_incident  
Disorder_incident  \
62110  E00023493              NaN                NaN  
```

```
1.0
42518  E00023692              NaN              NaN
NaN
35775  E00024133              NaN              NaN
NaN
32798  E00023762              NaN              NaN
1.0
42182  E00023845              NaN              NaN
1.0
46833  E00024057              NaN              NaN
NaN
24803  E00175193              NaN              NaN
NaN
22276  E00023936              NaN              NaN
1.0
10505  E00175201              1.0              NaN
NaN
52254  E00023648              NaN              NaN
NaN
13221  E00023711              NaN              1.0
NaN
38492  E00023502              NaN              NaN
NaN
58288  E00023534              NaN              NaN
NaN
10280  E00023936              1.0              NaN
NaN
14214  E00014084              NaN              NaN
NaN
32925  E00023866              NaN              NaN
1.0
55980  E00023580              NaN              NaN
NaN
25109  E00023945              NaN              NaN
1.0
76553  E00023695              NaN              NaN
1.0
3251   E00023859              NaN              NaN
NaN
14697  E00023506              NaN              NaN
NaN
36444  E00175214              NaN              NaN
NaN
10992  E00023509              NaN              NaN
NaN
33113  E00024091              NaN              NaN
1.0
28762  E00024083              NaN              NaN
1.0
32769  E00175245              NaN              NaN
```

```
1.0
39007  E00023500              NaN                NaN
NaN
54405  E00175183              NaN                NaN
NaN
21444  E00023960              NaN                NaN
1.0
33069  E00024138              NaN                NaN
NaN
13335  E00023922              1.0                NaN
NaN
59588  E00023571              NaN                NaN
NaN
11223  E00024113              1.0                NaN
NaN
45931  E00023494              NaN                NaN
1.0
15318  E00023471              NaN                1.0
NaN
24498  E00024130              NaN                NaN
NaN
11558  E00024068              NaN                NaN
NaN
10299  E00023672              NaN                NaN
NaN
2850         NaN              NaN                NaN
NaN
31555  E00023447              NaN                NaN
1.0
78762  E00024129              NaN                NaN
1.0
50505  E00024105              NaN                NaN
NaN
51835  E00023929              NaN                NaN
1.0
41708  E00023957              NaN                NaN
1.0
3352   E00024113              NaN                NaN
1.0

       Fraud_incident  Robbery_incident  SexRelated_incident  \
62110             NaN               NaN                  NaN
42518             NaN               NaN                  NaN
35775             NaN               NaN                  NaN
32798             NaN               NaN                  NaN
42182             NaN               NaN                  NaN
46833             NaN               NaN                  NaN
24803             NaN               NaN                  NaN
22276             NaN               NaN                  NaN
10505             NaN               NaN                  NaN
```

```
52254             NaN               NaN                  NaN
13221             NaN               NaN                  NaN
38492             NaN               NaN                  NaN
58288             NaN               NaN                  NaN
10280             NaN               NaN                  NaN
14214             NaN               1.0                  NaN
32925             NaN               NaN                  NaN
55980             NaN               NaN                  NaN
25109             NaN               NaN                  NaN
76553             NaN               NaN                  NaN
3251              NaN               1.0                  NaN
14697             NaN               1.0                  NaN
36444             NaN               NaN                  NaN
10992             NaN               1.0                  NaN
33113             NaN               NaN                  NaN
28762             NaN               NaN                  NaN
32769             NaN               NaN                  NaN
39007             NaN               NaN                  NaN
54405             NaN               NaN                  NaN
21444             NaN               NaN                  NaN
33069             NaN               NaN                  NaN
13335             NaN               NaN                  NaN
59588             NaN               NaN                  NaN
11223             NaN               NaN                  NaN
45931             NaN               NaN                  NaN
15318             NaN               NaN                  NaN
24498             NaN               NaN                  NaN
11558             NaN               1.0                  NaN
10299             NaN               1.0                  NaN
2850              NaN               NaN                  1.0
31555             NaN               NaN                  NaN
78762             NaN               NaN                  NaN
50505             NaN               NaN                  NaN
51835             NaN               NaN                  NaN
41708             NaN               NaN                  NaN
3352              NaN               NaN                  NaN

       Violence_incident  WeaponPossession_incident
62110                NaN                        NaN
42518                1.0                        NaN
35775                1.0                        NaN
32798                NaN                        NaN
42182                NaN                        NaN
46833                1.0                        NaN
24803                1.0                        NaN
22276                NaN                        NaN
10505                NaN                        NaN
52254                1.0                        NaN
13221                NaN                        NaN
38492                1.0                        NaN
```

```
58288                1.0                     NaN
10280                NaN                     NaN
14214                NaN                     NaN
32925                NaN                     NaN
55980                1.0                     NaN
25109                NaN                     NaN
76553                NaN                     NaN
3251                 NaN                     NaN
14697                NaN                     NaN
36444                1.0                     NaN
10992                NaN                     NaN
33113                NaN                     NaN
28762                NaN                     NaN
32769                NaN                     NaN
39007                1.0                     NaN
54405                1.0                     NaN
21444                NaN                     NaN
33069                1.0                     NaN
13335                NaN                     NaN
59588                1.0                     NaN
11223                NaN                     NaN
45931                NaN                     NaN
15318                NaN                     NaN
24498                1.0                     NaN
11558                NaN                     NaN
10299                NaN                     NaN
2850                 NaN                     NaN
31555                NaN                     NaN
78762                NaN                     NaN
50505                1.0                     NaN
51835                NaN                     NaN
41708                NaN                     NaN
3352                 NaN                     NaN
```

```
df = Crime_all[Crime_all.duplicated(['OAs'], keep=False)]
df.info()
df.sample(45, random_state=10)
```

```
             OAs  Damage_incident  Burglary_incident
Disorder_incident  \
70969  E00023934              NaN                NaN
1.0
8580   E00023945              NaN                NaN
1.0
73496  E00023936              NaN                NaN
1.0
12137  E00023517              NaN                NaN
NaN
5561   E00023494              NaN                NaN
NaN
12675  E00175191              NaN                1.0
```

```
NaN
57283  E00023620              NaN               NaN
1.0
16474  E00023758              NaN               NaN
NaN
72312  E00023714              NaN               NaN
1.0
8047   E00024084              NaN               NaN
1.0
13724  E00023943              1.0               NaN
NaN
61488  E00023959              NaN               NaN
NaN
7731   E00023935              1.0               NaN
NaN
54015  E00023482              NaN               NaN
1.0
39854  E00023932              NaN               NaN
NaN
22604  E00023544              NaN               NaN
1.0
12743  E00023596              NaN               1.0
NaN
65934  E00023731              NaN               NaN
NaN
1746   E00023556              NaN               1.0
NaN
11773  E00023881              NaN               1.0
NaN
50489  E00023695              NaN               NaN
NaN
17442  E00023787              NaN               NaN
NaN
70386  E00024048              NaN               NaN
NaN
7771   E00023934              NaN               NaN
NaN
39276  E00023965              NaN               NaN
NaN
22603  E00175209              NaN               NaN
NaN
33985  E00024113              NaN               NaN
1.0
33015  E00023564              NaN               NaN
1.0
9879   E00175206              NaN               NaN
NaN
68148  E00023602              NaN               NaN
NaN
2329   E00175250              NaN               1.0
```

```
NaN
16216  E00175192              NaN                NaN
NaN
4368   E00023817              NaN                NaN
NaN
35302  E00175241              NaN                NaN
1.0
15852  E00024118              NaN                NaN
1.0
30452  E00023595              NaN                NaN
1.0
12807  E00023470              NaN                NaN
NaN
57846  E00175196              NaN                NaN
1.0
13244  E00024137              NaN                NaN
NaN
5721   E00024084              NaN                1.0
NaN
407    E00175194              NaN                NaN
NaN
49278  E00175214              NaN                NaN
1.0
68353  E00023502              NaN                NaN
NaN
57227        NaN              NaN                NaN
NaN
27583  E00175188              NaN                NaN
NaN

       Fraud_incident  Robbery_incident  SexRelated_incident  \
70969             NaN               NaN                  NaN
8580              NaN               NaN                  NaN
73496             NaN               NaN                  NaN
12137             NaN               NaN                  NaN
5561              NaN               1.0                  NaN
12675             NaN               NaN                  NaN
57283             NaN               NaN                  NaN
16474             NaN               NaN                  NaN
72312             NaN               NaN                  NaN
8047              NaN               NaN                  NaN
13724             NaN               NaN                  NaN
61488             NaN               NaN                  NaN
7731              NaN               NaN                  NaN
54015             NaN               NaN                  NaN
39854             NaN               NaN                  NaN
22604             NaN               NaN                  NaN
12743             NaN               NaN                  NaN
65934             NaN               NaN                  NaN
1746              NaN               NaN                  NaN
```

```
11773             NaN               NaN                  NaN
50489             NaN               NaN                  NaN
17442             NaN               NaN                  NaN
70386             NaN               NaN                  NaN
7771              NaN               NaN                  NaN
39276             NaN               NaN                  NaN
22603             NaN               NaN                  NaN
33985             NaN               NaN                  NaN
33015             NaN               NaN                  NaN
9879              NaN               NaN                  NaN
68148             NaN               NaN                  NaN
2329              NaN               NaN                  NaN
16216             NaN               1.0                  NaN
4368              NaN               1.0                  NaN
35302             NaN               NaN                  NaN
15852             NaN               NaN                  NaN
30452             NaN               NaN                  NaN
12807             NaN               1.0                  NaN
57846             NaN               NaN                  NaN
13244             NaN               1.0                  NaN
5721              NaN               NaN                  NaN
407               NaN               1.0                  NaN
49278             NaN               NaN                  NaN
68353             NaN               NaN                  NaN
57227             NaN               NaN                  NaN
27583             NaN               NaN                  NaN

       Violence_incident  WeaponPossession_incident
70969                NaN                        NaN
8580                 NaN                        NaN
73496                NaN                        NaN
12137                1.0                        NaN
5561                 NaN                        NaN
12675                NaN                        NaN
57283                NaN                        NaN
16474                1.0                        NaN
72312                NaN                        NaN
8047                 NaN                        NaN
13724                NaN                        NaN
61488                1.0                        NaN
7731                 NaN                        NaN
54015                NaN                        NaN
39854                1.0                        NaN
22604                NaN                        NaN
12743                NaN                        NaN
65934                1.0                        NaN
1746                 NaN                        NaN
11773                NaN                        NaN
50489                1.0                        NaN
17442                1.0                        NaN
```

```
70386                1.0                        NaN
7771                 1.0                        NaN
39276                1.0                        NaN
22603                1.0                        NaN
33985                NaN                        NaN
33015                NaN                        NaN
9879                 1.0                        NaN
68148                1.0                        NaN
2329                 NaN                        NaN
16216                NaN                        NaN
4368                 NaN                        NaN
35302                NaN                        NaN
15852                NaN                        NaN
30452                NaN                        NaN
12807                NaN                        NaN
57846                NaN                        NaN
13244                NaN                        NaN
5721                 NaN                        NaN
407                  NaN                        NaN
49278                NaN                        NaN
68353                1.0                        NaN
57227                1.0                        NaN
27583                1.0                        NaN
```

```
df2=Crime_all.groupby(by=['OAs'], dropna=True).sum()
print(df2)
df2.info()
```

```
           Damage_incident  Burglary_incident  Disorder_incident  \
OAs
E00004172              3.0                4.0               46.0
E00004197              0.0                0.0                0.0
E00004201             13.0                0.0                4.0
E00004527             20.0                3.0              258.0
E00004609              0.0                0.0                0.0
...                    ...                ...                ...
E00175275             13.0                6.0              276.0
E00175276              3.0               11.0                3.0
E00175277             37.0               36.0              319.0
E00175278             19.0                8.0               65.0
E00175279              0.0                0.0                0.0

           Fraud_incident  Robbery_incident  SexRelated_incident  \
OAs
E00004172             0.0               4.0                  0.0
E00004197             0.0               0.0                  0.0
E00004201             0.0               6.0                  0.0
E00004527             0.0               1.0                  0.0
E00004609             0.0               0.0                  0.0
...                   ...               ...                  ...
```

```
E00175275             0.0               2.0                  0.0
E00175276             1.0               2.0                  0.0
E00175277             0.0              40.0                  0.0
E00175278             1.0               1.0                  0.0
E00175279             0.0               0.0                  0.0

           Violence_incident  WeaponPossession_incident
OAs
E00004172                5.0                        0.0
E00004197               11.0                        0.0
E00004201               10.0                        0.0
E00004527               56.0                        1.0
E00004609                2.0                        0.0
...                      ...                        ...
E00175275               72.0                        3.0
E00175276               14.0                        0.0
E00175277              183.0                        4.0
E00175278               42.0                        0.0
E00175279                4.0                        0.0

[746 rows x 8 columns]
<class 'lux.core.frame.LuxDataFrame'>
Index: 746 entries, E00004172 to E00175279
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Damage_incident            746 non-null    float64
 1   Burglary_incident          746 non-null    float64
 2   Disorder_incident          746 non-null    float64
 3   Fraud_incident             746 non-null    float64
 4   Robbery_incident           746 non-null    float64
 5   SexRelated_incident        746 non-null    float64
 6   Violence_incident          746 non-null    float64
 7   WeaponPossession_incident  746 non-null    float64
dtypes: float64(8)
memory usage: 52.5+ KB
```

```
df2.sample(45, random_state=10)
```

```
           Damage_incident  Burglary_incident  Disorder_incident  \
OAs
E00023657             13.0               22.0               35.0
E00023636             23.0               38.0              123.0
E00024101              0.0                0.0                1.0
E00023812              1.0               13.0                0.0
E00175190            125.0              202.0             1591.0
E00023705              0.0                0.0                0.0
E00023968              8.0               14.0                2.0
E00175225             44.0               39.0               96.0
E00024116             34.0               63.0              109.0
```

```
E00023840             11.0               56.0               61.0
E00023787             13.0               16.0               85.0
E00023989              0.0                0.0                0.0
E00023712              0.0                0.0                0.0
E00024068              4.0               10.0               29.0
E00023713              8.0               41.0               10.0
E00023428              0.0                0.0                0.0
E00023630              4.0               36.0              104.0
E00023851             25.0               13.0              196.0
E00023841             11.0               50.0               43.0
E00024036              7.0               17.0                3.0
E00023883             15.0                4.0               67.0
E00023551             15.0               14.0               40.0
E00023873             21.0               32.0              156.0
E00023781              5.0               27.0               55.0
E00023621              9.0               19.0               21.0
E00175259              3.0               14.0               48.0
E00023885              1.0                6.0                9.0
E00023878              0.0                0.0                0.0
E00024045              7.0                8.0               12.0
E00023445              9.0               22.0                7.0
E00023854             26.0               41.0              161.0
E00023468              6.0               17.0               65.0
E00023577             22.0               13.0              171.0
E00023876              4.0                5.0               14.0
E00024078              0.0                0.0                0.0
E00023726              2.0               22.0                1.0
E00023447             15.0               11.0               44.0
E00024064             32.0               27.0              285.0
E00175263             14.0                6.0               21.0
E00023938             45.0               22.0               65.0
E00023655              3.0                8.0                0.0
E00175262              0.0                0.0                0.0
E00023575             48.0               39.0              391.0
E00023666              1.0               13.0               16.0
E00023597              0.0                0.0                0.0

           Fraud_incident  Robbery_incident  SexRelated_incident  \
OAs
E00023657             0.0              16.0                  0.0
E00023636             9.0              17.0                  6.0
E00024101             0.0               0.0                  0.0
E00023812             0.0               0.0                  0.0
E00175190           141.0             703.0                118.0
E00023705             0.0               0.0                  0.0
E00023968             0.0               3.0                  0.0
E00175225             0.0               7.0                  0.0
E00024116             0.0              32.0                  0.0
E00023840             1.0              28.0                  0.0
E00023787             0.0              19.0                  0.0
```

```
E00023989             0.0               0.0                  0.0
E00023712             0.0               0.0                  0.0
E00024068             0.0               7.0                  0.0
E00023713             0.0               7.0                  0.0
E00023428             0.0               0.0                  0.0
E00023630             0.0               4.0                  0.0
E00023851             0.0               6.0                  0.0
E00023841             0.0              18.0                  0.0
E00024036             0.0               1.0                  1.0
E00023883             1.0               6.0                  0.0
E00023551             0.0               3.0                  0.0
E00023873             0.0              12.0                  0.0
E00023781             0.0               4.0                  0.0
E00023621             0.0               7.0                  0.0
E00175259             0.0               0.0                  0.0
E00023885             0.0               3.0                  0.0
E00023878             0.0               0.0                  0.0
E00024045             2.0               0.0                  0.0
E00023445             0.0               9.0                  0.0
E00023854             0.0               6.0                  0.0
E00023468             1.0               4.0                  0.0
E00023577             0.0              10.0                  0.0
E00023876             0.0               3.0                  0.0
E00024078             0.0               0.0                  0.0
E00023726             0.0               4.0                  0.0
E00023447             0.0               0.0                  0.0
E00024064             0.0              14.0                  0.0
E00175263             0.0               1.0                  0.0
E00023938             0.0              23.0                  1.0
E00023655             0.0               1.0                  0.0
E00175262             0.0               0.0                  0.0
E00023575             7.0              15.0                  6.0
E00023666             0.0               2.0                  0.0
E00023597             0.0               0.0                  1.0

           Violence_incident  WeaponPossession_incident
OAs
E00023657               33.0                        2.0
E00023636               84.0                        0.0
E00024101                8.0                        0.0
E00023812                7.0                        0.0
E00175190             1248.0                       50.0
E00023705                1.0                        0.0
E00023968               17.0                        0.0
E00175225               91.0                        0.0
E00024116               99.0                        2.0
E00023840               86.0                        2.0
E00023787              129.0                        0.0
E00023989                2.0                        0.0
E00023712                1.0                        0.0
```

```
E00024068               13.0                        1.0
E00023713               67.0                        2.0
E00023428                1.0                        0.0
E00023630               75.0                        0.0
E00023851               69.0                        4.0
E00023841               28.0                        0.0
E00024036               71.0                        0.0
E00023883               60.0                        0.0
E00023551               66.0                        0.0
E00023873               63.0                        3.0
E00023781                8.0                        0.0
E00023621               30.0                        3.0
E00175259               20.0                        0.0
E00023885                9.0                        0.0
E00023878                8.0                        0.0
E00024045               42.0                        0.0
E00023445               19.0                        0.0
E00023854               72.0                        1.0
E00023468               27.0                        0.0
E00023577              122.0                        1.0
E00023876               16.0                        1.0
E00024078                1.0                        0.0
E00023726               11.0                        0.0
E00023447               18.0                        0.0
E00024064               99.0                        3.0
E00175263               58.0                        3.0
E00023938               88.0                       33.0
E00023655               13.0                        0.0
E00175262                2.0                        0.0
E00023575              203.0                        6.0
E00023666               32.0                        0.0
E00023597                7.0                        0.0
```

```
Crime_all.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 209196 entries, 0 to 2197
Data columns (total 9 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   OAs                        204572 non-null  object
 1   Damage_incident            14077 non-null   float64
 2   Burglary_incident          19994 non-null   float64
 3   Disorder_incident          78977 non-null   float64
 4   Fraud_incident             1960 non-null    float64
 5   Robbery_incident           16828 non-null   float64
 6   SexRelated_incident        3548 non-null    float64
 7   Violence_incident          71614 non-null   float64
 8   WeaponPossession_incident  2198 non-null    float64
dtypes: float64(8), object(1)
memory usage: 16.0+ MB
```

```
Crime_all["Crime_all"] = " "
```

```
Crime_all['Crime_all'] = Crime_all['Damage_incident'] +
Crime_all['Burglary_incident'] + Crime_all['Disorder_incident'] +
Crime_all['Fraud_incident'] + Crime_all['Robbery_incident'] +
Crime_all['SexRelated_incident'] + Crime_all['Violence_incident'] +
Crime_all['WeaponPossession_incident']
```

```
Crime_all['Crime_all'] =
Crime_all[list(Crime_all.columns)].sum(axis=1)
```

```
/var/folders/c3/qc9fdqbj0lvdcbr2l68p4t2h0000gn/T/
ipykernel_1449/278294237.py:1: FutureWarning:Dropping of nuisance
columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError.  Select
only valid columns before calling the reduction.
```

```
Crime_all.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 209196 entries, 0 to 2197
Data columns (total 10 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   OAs                        204572 non-null  object
 1   Damage_incident            14077 non-null   float64
 2   Burglary_incident          19994 non-null   float64
 3   Disorder_incident          78977 non-null   float64
 4   Fraud_incident             1960 non-null    float64
 5   Robbery_incident           16828 non-null   float64
 6   SexRelated_incident        3548 non-null    float64
 7   Violence_incident          71614 non-null   float64
 8   WeaponPossession_incident  2198 non-null    float64
 9   Crime_all                  209196 non-null  float64
dtypes: float64(9), object(1)
memory usage: 17.6+ MB
```

```
Crime_all.sample(10)
```

```
             OAs  Damage_incident  Burglary_incident
Disorder_incident  \
59424  E00023814              NaN                NaN
NaN
28278  E00023935              NaN                NaN
NaN
6739   E00024052              NaN                NaN
1.0
31790  E00024118              NaN                NaN
1.0
31344  E00023893              NaN                NaN
NaN
18333  E00023937              NaN                NaN
```

```
NaN
13403  E00024114              1.0                NaN
NaN
11218  E00023842              NaN                NaN
NaN
6098   E00023510              1.0                NaN
NaN
49534  E00023935              NaN                NaN
NaN

       Fraud_incident  Robbery_incident  SexRelated_incident  \
59424             NaN               NaN                  NaN
28278             NaN               NaN                  NaN
6739              NaN               NaN                  NaN
31790             NaN               NaN                  NaN
31344             NaN               NaN                  NaN
18333             NaN               NaN                  NaN
13403             NaN               NaN                  NaN
11218             NaN               1.0                  NaN
6098              NaN               NaN                  NaN
49534             NaN               NaN                  NaN

       Violence_incident  WeaponPossession_incident  Crime_all
59424                1.0                        NaN        1.0
28278                1.0                        NaN        1.0
6739                 NaN                        NaN        1.0
31790                NaN                        NaN        1.0
31344                1.0                        NaN        1.0
18333                1.0                        NaN        1.0
13403                NaN                        NaN        1.0
11218                NaN                        NaN        1.0
6098                 NaN                        NaN        1.0
49534                1.0                        NaN        1.0
```

```
Crime_all.to_csv('Crime.csv', encoding='utf-8', index=False)
```

Noise and odour

```
Noise_odour.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Noise Complaint Index           90798 non-null  object
 1   Time                            90798 non-null  object
 2   Received Date                   90798 non-null  datetime64[ns]
 3   Financial Year                  90798 non-null  object
 4   Service Request                 90798 non-null  object
 5   Service Request Group Type      90798 non-null  object
 6   Service Request Group Sub Type  90798 non-null  object
 7   Service Request_1               90798 non-null  object
 8   Address Key                     90798 non-null  object
 9   Type of Address                 90798 non-null  object
 10  Output Area Code (2011)         90798 non-null  object
 11  MSOA 2011 Code                  90798 non-null  object
 12  MSOA 2011 Name                  90798 non-null  object
 13  LSOA 2011 Code                  90798 non-null  object
 14  LSOA 2011 Name                  90798 non-null  object
 15  Ward Name                       90798 non-null  object
dtypes: datetime64[ns](1), object(15)
memory usage: 11.1+ MB
```

```
Noise = Noise_odour.drop(columns=['Noise Complaint Index', 'Time',
'Received Date', 'Financial Year', 'Service Request', 'Service Request
Group Type', 'Service Request_1', 'Address Key ', 'Type of Address',
'MSOA 2011 Code', 'MSOA 2011 Name', 'LSOA 2011 Code', 'LSOA 2011
Name', 'Ward Name'])
Noise.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 2 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Service Request Group Sub Type  90798 non-null  object
 1   Output Area Code (2011)         90798 non-null  object
dtypes: object(2)
memory usage: 1.4+ MB
```

```
Noise = Noise.rename(columns={'Output Area Code (2011)': 'OAs'})
```

```
Noise.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 2 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Service Request Group Sub Type  90798 non-null  object
 1   OAs                             90798 non-null  object
dtypes: object(2)
memory usage: 1.4+ MB
```

```
Noise['Incident'] = " "
```

```
Noise['Incident'] = 1
```

```
Noise["Animal"] = " "
Noise["Building Site"] = " "
Noise["Commercial Premises"] = " "
Noise["Email Complaint (1d)"] = " "
```

```
Noise["Formal complaints"] = " "
Noise["Non Noise Complaint (45m)"] = " "
Noise["Non Noise Complaint (4d)"] = " "
Noise["Proactive Noise"] = " "
Noise["Property Alarm"] = " "
Noise["Residential Premises"] = " "
Noise["Street"] = " "
Noise["VIP complaint"] = " "

Noise.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 15 columns):
 #   Column                          Non-Null Count  Dtype 
---  ------                          --------------  ----- 
 0   Service Request Group Sub Type  90798 non-null  object
 1   OAs                             90798 non-null  object
 2   Incident                        90798 non-null  int64 
 3   Animal                          90798 non-null  object
 4   Building Site                   90798 non-null  object
 5   Commercial Premises             90798 non-null  object
 6   Email Complaint (1d)            90798 non-null  object
 7   Formal complaints               90798 non-null  object
 8   Non Noise Complaint (45m)       90798 non-null  object
 9   Non Noise Complaint (4d)        90798 non-null  object
 10  Proactive Noise                 90798 non-null  object
 11  Property Alarm                  90798 non-null  object
 12  Residential Premises            90798 non-null  object
 13  Street                          90798 non-null  object
 14  VIP complaint                   90798 non-null  object
dtypes: int64(1), object(14)
memory usage: 10.4+ MB
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 
'Animal' :
        Noise.loc[index, 'Animal'] = '1'

    else:
        Noise.loc[index, 'Animal'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
2819                          Street  E00023931         1      0   
12599                         Street  E00023814         1      0   
43433           Residential Premises  E00023733         1      0   
43762           Residential Premises  E00023578         1      0   
85705                         Animal  E00024021         1      1   
```

```
8255            Residential Premises  E00023826         1      0
66482           Residential Premises  E00023439         1      0
87221                         Street  E00175247         1      0
83470                         Street  E00023926         1      0
46449                         Street  E00023839         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
2819
12599
43433
43762
85705
8255
66482
87221
83470
46449

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
2819

12599

43433

43762

85705

8255

66482

87221

83470

46449


      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
2819

12599

43433
```

```
43762

85705

8255

66482

87221

83470

46449
```

```python
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 'Building
Site' :
        Noise.loc[index, 'Building Site'] = '1'

    else:
        Noise.loc[index, 'Building Site'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
50433           Residential Premises  E00023987         1      0
38578                         Street  E00175206         1      0
13050           Residential Premises  E00024024         1      0
23693      Non Noise Complaint (45m)  E00024130         1      0
51452           Residential Premises  E00175188         1      0
15079           Residential Premises  E00024002         1      0
24246           Residential Premises  E00023530         1      0
57465           Residential Premises  E00023694         1      0
62489           Residential Premises  E00023813         1      0
52564           Residential Premises  E00024125         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
50433             0
38578             0
13050             0
23693             0
51452             0
15079             0
24246             0
57465             0
62489             0
52564             0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
```

```
(4d)  \
50433

38578

13050

23693

51452

15079

24246

57465

62489

52564


      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
50433

38578

13050

23693

51452

15079

24246

57465

62489

52564
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] ==
'Commercial Premises' :
        Noise.loc[index, 'Commercial Premises'] = '1'
    else:
        Noise.loc[index, 'Commercial Premises'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
7737             Commercial Premises  E00023824         1      0
16769            Commercial Premises  E00023936         1      0
48600           Residential Premises  E00023722         1      0
45878                  Building Site  E00023883         1      0
37072           Residential Premises  E00024070         1      0
85669           Residential Premises  E00023494         1      0
55225           Residential Premises  E00024133         1      0
23698           Residential Premises  E00024012         1      0
56456           Residential Premises  E00024126         1      0
63516           Residential Premises  E00023540         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
7737              0                   1
16769             0                   1
48600             0                   0
45878             1                   0
37072             0                   0
85669             0                   0
55225             0                   0
23698             0                   0
56456             0                   0
63516             0                   0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
7737

16769

48600

45878

37072

85669

55225

23698
```

```
56456

63516


      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
7737

16769

48600

45878

37072

85669

55225

23698

56456

63516
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 'Email
Complaint (1d)' :
        Noise.loc[index, 'Email Complaint (1d)'] = '1'

    else:
        Noise.loc[index, 'Email Complaint (1d)'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
88593           Residential Premises  E00023832         1      0
23005                         Animal  E00024085         1      1
33094                  Building Site  E00023476         1      0
5206                          Street  E00023422         1      0
58677           Residential Premises  E00023798         1      0
83071           Residential Premises  E00024126         1      0
43027           Residential Premises  E00024024         1      0
88826                  Building Site  E00023839         1      0
7260                          Street  E00023830         1      0
79613            Commercial Premises  E00023627         1      0
```

```
      Building Site Commercial Premises Email Complaint (1d)  \
88593             0                   0                    0
23005             0                   0                    0
33094             1                   0                    0
5206              0                   0                    0
58677             0                   0                    0
83071             0                   0                    0
43027             0                   0                    0
88826             1                   0                    0
7260              0                   0                    0
79613             0                   1                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
88593

23005

33094

5206

58677

83071

43027

88826

7260

79613


      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
88593

23005

33094

5206

58677
```

```
83071

43027

88826

7260

79613
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 'Formal
complaints' :
        Noise.loc[index, 'Formal complaints'] = '1'

    else:
        Noise.loc[index, 'Formal complaints'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
49649           Residential Premises  E00023931         1      0
3882                   Building Site  E00023930         1      0
75915                  Building Site  E00023561         1      0
34372           Residential Premises  E00023712         1      0
53755            Commercial Premises  E00023980         1      0
5796                          Street  E00023863         1      0
17802           Residential Premises  E00023684         1      0
88025           Residential Premises  E00024125         1      0
64853                 Property Alarm  E00024130         1      0
29267           Residential Premises  E00024126         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
49649             0                   0                    0
3882              1                   0                    0
75915             1                   0                    0
34372             0                   0                    0
53755             0                   1                    0
5796              0                   0                    0
17802             0                   0                    0
88025             0                   0                    0
64853             0                   0                    0
29267             0                   0                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
49649                 0

3882                  0
```

```
75915                 0

34372                 0

53755                 0

5796                  0

17802                 0

88025                 0

64853                 0

29267                 0


      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
49649

3882

75915

34372

53755

5796

17802

88025

64853

29267
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 'Non
Noise Complaint (45m)' :
        Noise.loc[index, 'Non Noise Complaint (45m)'] = '1'

    else:
        Noise.loc[index, 'Non Noise Complaint (45m)'] = '0'
Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
52530            Commercial Premises  E00175183         1      0
51312           Residential Premises  E00024095         1      0
11279           Residential Premises  E00023516         1      0
3741        Non Noise Complaint (4d)  E00023992         1      0
36160            Commercial Premises  E00175187         1      0
6488                          Street  E00024041         1      0
9243            Residential Premises  E00023784         1      0
63765                 Property Alarm  E00023550         1      0
70007                         Street  E00175192         1      0
59583           Residential Premises  E00024122         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
52530             0                   1                    0
51312             0                   0                    0
11279             0                   0                    0
3741              0                   0                    0
36160             0                   1                    0
6488              0                   0                    0
9243              0                   0                    0
63765             0                   0                    0
70007             0                   0                    0
59583             0                   0                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
52530                 0                         0

51312                 0                         0

11279                 0                         0

3741                  0                         0

36160                 0                         0

6488                  0                         0

9243                  0                         0

63765                 0                         0

70007                 0                         0

59583                 0                         0
```

```
      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
52530

51312

11279

3741

36160

6488

9243

63765

70007

59583
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 'Non
Noise Complaint (4d)' :
        Noise.loc[index, 'Non Noise Complaint (4d)'] = '1'

    else:
        Noise.loc[index, 'Non Noise Complaint (4d)'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
53826      Non Noise Complaint (45m)  E00023494         1      0
30973                  Building Site  E00024135         1      0
8796            Residential Premises  E00023429         1      0
14297            Commercial Premises  E00023929         1      0
70432           Residential Premises  E00024061         1      0
33953           Residential Premises  E00023943         1      0
1821            Residential Premises  E00175198         1      0
86991           Residential Premises  E00024017         1      0
82179            Commercial Premises  E00023453         1      0
31189           Residential Premises  E00024108         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
53826             0                   0                    0
30973             1                   0                    0
```

```
8796              0                   0                    0
14297             0                   1                    0
70432             0                   0                    0
33953             0                   0                    0
1821              0                   0                    0
86991             0                   0                    0
82179             0                   1                    0
31189             0                   0                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
53826                 0                         1
0
30973                 0                         0
0
8796                  0                         0
0
14297                 0                         0
0
70432                 0                         0
0
33953                 0                         0
0
1821                  0                         0
0
86991                 0                         0
0
82179                 0                         0
0
31189                 0                         0
0

      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
53826

30973

8796

14297

70432

33953

1821

86991
```

```
82179

31189
```

```python
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] ==
'Proactive Noise' :
        Noise.loc[index, 'Proactive Noise'] = '1'

    else:
        Noise.loc[index, 'Proactive Noise'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
13677           Residential Premises  E00023544         1      0
15524                         Street  E00023945         1      0
58419           Residential Premises  E00023979         1      0
17025                         Street  E00175194         1      0
27148           Residential Premises  E00023706         1      0
68855           Residential Premises  E00023538         1      0
54689                         Street  E00023845         1      0
53090           Residential Premises  E00023574         1      0
67722           Residential Premises  E00023889         1      0
34518           Residential Premises  E00023463         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
13677             0                   0                    0
15524             0                   0                    0
58419             0                   0                    0
17025             0                   0                    0
27148             0                   0                    0
68855             0                   0                    0
54689             0                   0                    0
53090             0                   0                    0
67722             0                   0                    0
34518             0                   0                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
13677                 0                         0
0
15524                 0                         0
0
58419                 0                         0
0
17025                 0                         0
0
```

```
27148                 0                         0
0
68855                 0                         0
0
54689                 0                         0
0
53090                 0                         0
0
67722                 0                         0
0
34518                 0                         0
0

      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
13677               0

15524               0

58419               0

17025               0

27148               0

68855               0

54689               0

53090               0

67722               0

34518               0
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 'Property
Alarm' :
        Noise.loc[index, 'Property Alarm'] = '1'

    else:
        Noise.loc[index, 'Property Alarm'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
9614                   Building Site  E00023887         1      0
33664            Commercial Premises  E00174651         1      0
39777                         Street  E00175194         1      0
```

```
54346           Residential Premises  E00175226         1      0
73833       Non Noise Complaint (4d)  E00023466         1      0
42135           Residential Premises  E00023790         1      0
49222           Residential Premises  E00175251         1      0
9948                   Building Site  E00024128         1      0
29550                         Street  E00023945         1      0
66565           Residential Premises  E00023826         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
9614              1                   0                    0
33664             0                   1                    0
39777             0                   0                    0
54346             0                   0                    0
73833             0                   0                    0
42135             0                   0                    0
49222             0                   0                    0
9948              1                   0                    0
29550             0                   0                    0
66565             0                   0                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
9614                  0                         0
0
33664                 0                         0
0
39777                 0                         0
0
54346                 0                         0
0
73833                 0                         0
1
42135                 0                         0
0
49222                 0                         0
0
9948                  0                         0
0
29550                 0                         0
0
66565                 0                         0
0

      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
9614                0              0

33664               0              0

39777               0              0
```

```
54346               0              0

73833               0              0

42135               0              0

49222               0              0

9948                0              0

29550               0              0

66565               0              0
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] ==
'Street' :
        Noise.loc[index, 'Street'] = '1'

    else:
        Noise.loc[index, 'Street'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
64633           Residential Premises  E00175273         1      0
50652            Commercial Premises  E00175191         1      0
6438                   Building Site  E00023453         1      0
18002           Residential Premises  E00023484         1      0
6380             Commercial Premises  E00024136         1      0
81191           Residential Premises  E00023652         1      0
10976                         Street  E00023517         1      0
48787                         Street  E00023517         1      0
62055           Residential Premises  E00023937         1      0
59083                 Property Alarm  E00175190         1      0

      Building Site Commercial Premises Email Complaint (1d)  \
64633             0                   0                    0
50652             0                   1                    0
6438              1                   0                    0
18002             0                   0                    0
6380              0                   1                    0
81191             0                   0                    0
10976             0                   0                    0
48787             0                   0                    0
62055             0                   0                    0
59083             0                   0                    0
```

```
      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
64633                 0                         0
0
50652                 0                         0
0
6438                  0                         0
0
18002                 0                         0
0
6380                  0                         0
0
81191                 0                         0
0
10976                 0                         0
0
48787                 0                         0
0
62055                 0                         0
0
59083                 0                         0
0

      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
64633               0              0                           0

50652               0              0                           0

6438                0              0                           0

18002               0              0                           0

6380                0              0                           0

81191               0              0                           0

10976               0              0                           1

48787               0              0                           1

62055               0              0                           0

59083               0              1                           0
```

```python
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] == 'VIP
complaint' :
        Noise.loc[index, 'VIP complaint'] = '1'
    else:
        Noise.loc[index, 'VIP complaint'] = '0'

Noise.info()
Noise.sample(10)

      Service Request Group Sub Type        OAs  Incident Animal  \
25057           Residential Premises  E00175272        1      0
32762           Residential Premises  E00023586        1      0
6265            Residential Premises  E00023872        1      0
25122           Residential Premises  E00023596        1      0
15423                  Building Site  E00175249        1      0
58021           Residential Premises  E00175226        1      0
86501           Residential Premises  E00023709        1      0
41659           Residential Premises  E00175239        1      0
78526                         Street  E00023957        1      0
26206                  Building Site  E00024096        1      0

      Building Site Commercial Premises Email Complaint (1d)  \
25057             0                   0                    0
32762             0                   0                    0
6265              0                   0                    0
25122             0                   0                    0
15423             1                   0                    0
58021             0                   0                    0
86501             0                   0                    0
41659             0                   0                    0
78526             0                   0                    0
26206             1                   0                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
25057                 0                         0
0
32762                 0                         0
0
6265                  0                         0
0
25122                 0                         0
0
15423                 0                         0
0
58021                 0                         0
0
86501                 0                         0
0
41659                 0                         0
0
78526                 0                         0
```

```
0
26206                 0                         0
0

      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
25057               0              0                           0
0
32762               0              0                           0
0
6265                0              0                           0
0
25122               0              0                           0
0
15423               0              0                           0
0
58021               0              0                           0
0
86501               0              0                           0
0
41659               0              0                           0
0
78526               0              0                           1
0
26206               0              0                           0
0
```

```
for index, row in Noise.iterrows() :
    if Noise.loc[index, 'Service Request Group Sub Type'] ==
'Residential Premises' :
        Noise.loc[index, 'Residential Premises'] = '1'

    else:
        Noise.loc[index, 'Residential Premises'] = '0'

Noise.info()
Noise.sample(10)
```

```
      Service Request Group Sub Type        OAs  Incident Animal  \
841              Commercial Premises  E00023998         1      0
45613                         Street  E00023450         1      0
34656                         Street  E00023560         1      0
59270           Residential Premises  E00023680         1      0
61159           Residential Premises  E00023460         1      0
53270            Commercial Premises  E00024136         1      0
43434           Residential Premises  E00175222         1      0
39655                         Street  E00023839         1      0
22782           Residential Premises  E00024078         1      0
67353                         Street  E00023602         1      0
```

```
      Building Site Commercial Premises Email Complaint (1d)  \
841               0                   1                    0
45613             0                   0                    0
34656             0                   0                    0
59270             0                   0                    0
61159             0                   0                    0
53270             0                   1                    0
43434             0                   0                    0
39655             0                   0                    0
22782             0                   0                    0
67353             0                   0                    0

      Formal complaints Non Noise Complaint (45m) Non Noise Complaint
(4d)  \
841                   0                         0
0
45613                 0                         0
0
34656                 0                         0
0
59270                 0                         0
0
61159                 0                         0
0
53270                 0                         0
0
43434                 0                         0
0
39655                 0                         0
0
22782                 0                         0
0
67353                 0                         0
0

      Proactive Noise Property Alarm Residential Premises Street VIP
complaint
841                 0              0                    0      0
0
45613               0              0                    0      1
0
34656               0              0                    0      1
0
59270               0              0                    1      0
0
61159               0              0                    1      0
0
53270               0              0                    0      0
0
43434               0              0                    1      0
```

```
0
39655               0               0                   0      1
0
22782               0               0                   1      0
0
67353               0               0                   0      1
0
```

```
Noise.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 15 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Service Request Group Sub Type  90798 non-null  object
 1   OAs                             90798 non-null  object
 2   Incident                        90798 non-null  int64
 3   Animal                          90798 non-null  object
 4   Building Site                   90798 non-null  object
 5   Commercial Premises             90798 non-null  object
 6   Email Complaint (1d)            90798 non-null  object
 7   Formal complaints               90798 non-null  object
 8   Non Noise Complaint (45m)       90798 non-null  object
 9   Non Noise Complaint (4d)        90798 non-null  object
 10  Proactive Noise                 90798 non-null  object
 11  Property Alarm                  90798 non-null  object
 12  Residential Premises            90798 non-null  object
 13  Street                          90798 non-null  object
 14  VIP complaint                   90798 non-null  object
dtypes: int64(1), object(14)
memory usage: 10.4+ MB
```

```
Noise2 = Noise.drop(columns=['Service Request Group Sub Type'])
Noise2.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   OAs                        90798 non-null  object
 1   Incident                   90798 non-null  int64
 2   Animal                     90798 non-null  object
 3   Building Site              90798 non-null  object
 4   Commercial Premises        90798 non-null  object
 5   Email Complaint (1d)       90798 non-null  object
 6   Formal complaints          90798 non-null  object
 7   Non Noise Complaint (45m)  90798 non-null  object
 8   Non Noise Complaint (4d)   90798 non-null  object
 9   Proactive Noise            90798 non-null  object
 10  Property Alarm             90798 non-null  object
 11  Residential Premises       90798 non-null  object
 12  Street                     90798 non-null  object
 13  VIP complaint              90798 non-null  object
dtypes: int64(1), object(13)
memory usage: 9.7+ MB
```

```
Noise2.sample(10)
```

```
             OAs  Incident Animal Building Site Commercial Premises  \
51417  E00023467         1      0             0                   0
46281  E00024108         1      0             0                   0
73892  E00023967         1      0             0                   0
28     E00023456         1      0             0                   0
61150  E00023633         1      0             1                   0
51300  E00023939         1      0             0                   0
83727  E00023990         1      0             0                   0
28649  E00024126         1      0             0                   0
66018  E00023763         1      0             0                   0
25810  E00023657         1      0             0                   0

      Email Complaint (1d) Formal complaints Non Noise Complaint (45m)
\
51417                    0                 0                         0

46281                    0                 0                         0

73892                    0                 0                         0

28                       0                 0                         0

61150                    0                 0                         0

51300                    0                 0                         0

83727                    0                 0                         0

28649                    0                 0                         0

66018                    0                 0                         0

25810                    0                 0                         0


      Non Noise Complaint (4d) Proactive Noise Property Alarm  \
51417                        0               0              0
46281                        0               0              0
73892                        0               0              0
28                           0               0              1
```

```
61150                        0               0              0
51300                        0               0              0
83727                        0               0              0
28649                        0               0              0
66018                        0               0              0
25810                        0               0              0

      Residential Premises Street VIP complaint
51417                    1      0             0
46281                    0      1             0
73892                    1      0             0
28                       0      0             0
61150                    0      0             0
51300                    0      1             0
83727                    1      0             0
28649                    1      0             0
66018                    1      0             0
25810                    1      0             0
```

```
Noise2["Animal"] = Noise2["Animal"].astype(float)
Noise2["Building Site"] = Noise2["Building Site"].astype(float)
Noise2["Commercial Premises"] = Noise2["Commercial
Premises"].astype(float)
Noise2["Email Complaint (1d)"] = Noise2["Email Complaint
(1d)"].astype(float)
Noise2["Formal complaints"] = Noise2["Formal
complaints"].astype(float)
Noise2["Non Noise Complaint (45m)"] = Noise2["Non Noise Complaint
(45m)"].astype(float)
Noise2["Non Noise Complaint (4d)"] = Noise2["Non Noise Complaint
(4d)"].astype(float)
Noise2["Proactive Noise"] = Noise2["Proactive Noise"].astype(float)
Noise2["Property Alarm"] = Noise2["Property Alarm"].astype(float)
Noise2["Residential Premises"] = Noise2["Residential
Premises"].astype(float)
Noise2["Street"] = Noise2["Street"].astype(float)
Noise2["VIP complaint"] = Noise2["VIP complaint"].astype(float)

Noise2.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   OAs                        90798 non-null  object
 1   Incident                   90798 non-null  int64
 2   Animal                     90798 non-null  float64
 3   Building Site              90798 non-null  float64
 4   Commercial Premises        90798 non-null  float64
 5   Email Complaint (1d)       90798 non-null  float64
 6   Formal complaints          90798 non-null  float64
 7   Non Noise Complaint (45m)  90798 non-null  float64
 8   Non Noise Complaint (4d)   90798 non-null  float64
 9   Proactive Noise            90798 non-null  float64
 10  Property Alarm             90798 non-null  float64
 11  Residential Premises       90798 non-null  float64
 12  Street                     90798 non-null  float64
 13  VIP complaint              90798 non-null  float64
dtypes: float64(12), int64(1), object(1)
memory usage: 9.7+ MB
```

```
Noise2.sample(10)
```

```
             OAs  Incident  Animal  Building Site  Commercial Premises  
\
58856  E00023562         1     0.0            0.0                  0.0   

78340  E00023595         1     0.0            0.0                  0.0   

59667  E00023575         1     0.0            0.0                  1.0   

4898   E00024117         1     0.0            0.0                  0.0   

21885  E00024125         1     0.0            1.0                  0.0   

32231  E00023927         1     0.0            0.0                  0.0   

17685  E00023937         1     0.0            0.0                  0.0   

52488  E00024126         1     0.0            0.0                  0.0   

76369  E00023941         1     0.0            0.0                  0.0   

33119  E00023917         1     0.0            0.0                  0.0   


       Email Complaint (1d)  Formal complaints  Non Noise Complaint 
(45m)  \
58856                   0.0                0.0                        
0.0   
78340                   0.0                0.0                        
0.0   
59667                   0.0                0.0                        
0.0   
4898                    0.0                0.0                        
0.0   
21885                   0.0                0.0                        
0.0   
32231                   0.0                0.0                        
```

```
0.0
17685                   0.0                0.0
0.0
52488                   0.0                0.0
0.0
76369                   0.0                0.0
0.0
33119                   0.0                0.0
0.0

       Non Noise Complaint (4d)  Proactive Noise  Property Alarm  \
58856                       0.0              0.0             0.0
78340                       0.0              0.0             1.0
59667                       0.0              0.0             0.0
4898                        0.0              0.0             0.0
21885                       0.0              0.0             0.0
32231                       0.0              0.0             1.0
17685                       0.0              0.0             0.0
52488                       0.0              0.0             0.0
76369                       0.0              0.0             0.0
33119                       0.0              0.0             1.0

       Residential Premises  Street  VIP complaint
58856                   1.0     0.0            0.0
78340                   0.0     0.0            0.0
59667                   0.0     0.0            0.0
4898                    1.0     0.0            0.0
21885                   0.0     0.0            0.0
32231                   0.0     0.0            0.0
17685                   0.0     1.0            0.0
52488                   1.0     0.0            0.0
76369                   0.0     1.0            0.0
33119                   0.0     0.0            0.0
```

```
Noise3 = Noise2.rename(columns={'Incident': 'Noise_IncidentALL'})
```

```
Noise3.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   OAs                        90798 non-null  object
 1   Noise_IncidentALL          90798 non-null  int64
 2   Animal                     90798 non-null  float64
 3   Building Site              90798 non-null  float64
 4   Commercial Premises        90798 non-null  float64
 5   Email Complaint (1d)       90798 non-null  float64
 6   Formal complaints          90798 non-null  float64
 7   Non Noise Complaint (45m)  90798 non-null  float64
 8   Non Noise Complaint (4d)   90798 non-null  float64
 9   Proactive Noise            90798 non-null  float64
 10  Property Alarm             90798 non-null  float64
 11  Residential Premises       90798 non-null  float64
 12  Street                     90798 non-null  float64
 13  VIP complaint              90798 non-null  float64
dtypes: float64(12), int64(1), object(1)
memory usage: 9.7+ MB
```

```
Noise3.to_csv('Noise.csv', encoding='utf-8', index=False)
```

Licensing Data

```
Licensing['STREET']= Licensing['STREET'].astype("string")
Licensing['ADDRESS']= Licensing['ADDRESS'].astype("string")
```

```
Licensing.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 249616 entries, 0 to 249615
Data columns (total 11 columns):
 #   Column    Non-Null Count   Dtype 
---  ------    --------------   ----- 
 0   REFVAL    249616 non-null  object
 1   ADDRESS   249616 non-null  string
 2   LIUSE     237886 non-null  object
 3   ISSUED    249616 non-null  object
 4   LIPERMIT  248905 non-null  object
 5   LICYCLE   224063 non-null  object
 6   FTYPE     249616 non-null  object
 7   FVALUE    232187 non-null  object
 8   OPENT     223901 non-null  object
 9   CLOST     223906 non-null  object
 10  STREET    249616 non-null  string
dtypes: object(9), string(2)
memory usage: 20.9+ MB
```

```
Licensing.replace(to_replace=[r"\\t|\\n|\\r", "\t|\n|\r"],
value=["",""], regex=True)
```

```
                 REFVAL                                          
ADDRESS  \
0        12/02725/LIPDPS                     21 Romilly StreetLondonW1D
5AF
1        13/04950/LIPDPS                       25 London StreetLondonW2
1HH
2         16/01994/LIPVM                      25 Sheldon SquareLondonW2
6EY
3         16/07041/LIPCH           Ground Floor5 Goslett YardLondonWC2H
0EE
4         16/06777/LIPVM      Royal Albert HallKensington GoreLondonSW7
2AP
```

```
...                 ...
...
249611  22/04156/LIPDPS     The Newman Arms23 Rathbone StreetLondonW1T
1NG
249612   22/04162/LIPVM                        39 Duke StreetLondonW1U
1LP
249613   22/04162/LIPVM                        39 Duke StreetLondonW1U
1LP
249614  22/04177/LIPDPS              Ham Yard Hotel1 Ham YardLondonW1D
7DT
249615   22/06124/LIPCH  Development Site At 1 - 4 Walker's Court And8
...

        LIUSE      ISSUED LIPERMIT LICYCLE    FTYPE  FVALUE     OPENT  
\
0       PT234  06/03/2018   LATENR   00Z14   EXEMPT      NO  23:00:00

1       PT227  30/08/3013    ENTML   00Z56   NUMBER   81000  10:00:00

2       RT234  18/06/2018    ENTMR    SBBH  ALCOHOL    YESS  10:00:00

3       PT199  23/03/2018   ENTOTH    00WD  ALCOHOL     YES  09:00:00

4       PT070  31/05/2018   ENTPLA   00ALL   EXEMPT      NO  09:15:00

...       ...         ...      ...     ...      ...     ...       ...

249611  PT226  10/05/2022   LATENR   00Z56   NUMBER  47,000  23:00:00

249612  PT226  06/05/2022   RETALC   00Z14   NUMBER    <100  10:00:00

249613  PT226  06/05/2022   RETALC   07SUN  ALCOHOL     YES  11:00:00

249614  PT138  13/05/2022    ENTMR   00ALL  ALCOHOL    YESS  00:00:00

249615  PT199  16/06/2022   ENTOTH   07SUN   NUMBER       0  12:00:00


           CLOST               STREET
0       23:30:00  StreetLondonW1D 5AF
1       00:00:00   StreetLondonW2 1HH
2       00:00:00   SquareLondonW2 6EY
3       03:00:00   YardLondonWC2H 0EE
4       01:00:00    GoreLondonSW7 2AP
...          ...                  ...
249611  00:00:00  StreetLondonW1T 1NG
249612  00:00:00  StreetLondonW1U 1LP
249613  22:30:00  StreetLondonW1U 1LP
249614  00:00:00    YardLondonW1D 7DT
```

```
249615  22:30:00  StreetLondonW1F 0SB

[249616 rows x 11 columns]
```

```
Licensing.sample(5, random_state=10)
```

```
                 REFVAL
ADDRESS  \
132116  20/00884/LIPDPS  Saw Swee Hock Centre
1 Sheffield Street
London...
137387  18/13729/LIPDPS                         210 Strand
London
WC2R 1AP
65543     20/03567/LIPV                  25 Albemarle Street
London
W1S 4HX
119562    18/00645/LIPN                  9 Fitzmaurice Place
London
W1J 5JD
90288   18/03030/LIPDPS            Stratton House
5 Stratton Street
London

        LIUSE      ISSUED LIPERMIT LICYCLE    FTYPE FVALUE
OPENT  \
132116  RT199  05/03/2020   ENTOTH   00Z56   EXEMPT     NO  09:00:00

137387  PT303  14/12/2018   LATENR   00Z56  ALCOHOL    YES  23:00:00

65543   PT234  13/04/2021   RETALC   07SUN   NUMBER    NaN  12:00:00

119562  RT061  02/04/2018   ENTIND   00ALL   EXEMPT     NO  07:00:00

90288   PT234  04/04/2018   RETALC   00Z14   NUMBER   <250  10:00:00


           CLOST                  STREET
132116  03:00:00  Street
London
WC2A 2AP
137387  00:00:00  Strand
London
WC2R 1AP
65543   22:30:00   Street
London
W1S 4HX
119562  01:00:00    Place
London
W1J 5JD
```

```
90288   23:30:00  Stratton Street
London
```

```
DF= Licensing.STREET.str.split(expand=True,)
print(DF)
```

```
             0       1     2    3     4
0       Street  London   W1D  5AF  <NA>
1       Street  London    W2  1HH  <NA>
2       Square  London    W2  6EY  <NA>
3         Yard  London  WC2H  0EE  <NA>
4         Gore  London   SW7  2AP  <NA>
...        ...     ...   ...  ...   ...
249611  Street  London   W1T  1NG  <NA>
249612  Street  London   W1U  1LP  <NA>
249613  Street  London   W1U  1LP  <NA>
249614    Yard  London   W1D  7DT  <NA>
249615  Street  London   W1F  0SB  <NA>

[249616 rows x 5 columns]
```

```
DF.columns=['Street', 'City', 'PostCode1', 'Postcode2', 'None']
#DF= DF.rename(columns = {'0':'1','1':'2','2':'3', '3':'4', '4':'5'})
DF.info()
print(DF)
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 249616 entries, 0 to 249615
Data columns (total 5 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   Street     249616 non-null  string
 1   City       247634 non-null  string
 2   PostCode1  247634 non-null  string
 3   Postcode2  244305 non-null  string
 4   None       3226 non-null    string
dtypes: string(5)
memory usage: 9.5 MB
        Street    City PostCode1 Postcode2  None
0       Street  London       W1D       5AF  <NA>
1       Street  London        W2       1HH  <NA>
2       Square  London        W2       6EY  <NA>
3         Yard  London      WC2H       0EE  <NA>
4         Gore  London       SW7       2AP  <NA>
...        ...     ...       ...       ...   ...
249611  Street  London       W1T       1NG  <NA>
249612  Street  London       W1U       1LP  <NA>
249613  Street  London       W1U       1LP  <NA>
249614    Yard  London       W1D       7DT  <NA>
249615  Street  London       W1F       0SB  <NA>
```

```
[249616 rows x 5 columns]
```

```
df = DF.drop(columns=['Street', 'City', 'None'])
print(df)
```

```
       PostCode1 Postcode2
0            W1D       5AF
1             W2       1HH
2             W2       6EY
3           WC2H       0EE
4            SW7       2AP
...          ...       ...
249611       W1T       1NG
249612       W1U       1LP
249613       W1U       1LP
249614       W1D       7DT
249615       W1F       0SB

[249616 rows x 2 columns]
```

```
df['Postcode']= df['PostCode1'] + df['Postcode2']
print(df)
```

```
       PostCode1 Postcode2 Postcode
0            W1D       5AF   W1D5AF
1             W2       1HH    W21HH
2             W2       6EY    W26EY
3           WC2H       0EE  WC2H0EE
4            SW7       2AP   SW72AP
...          ...       ...      ...
249611       W1T       1NG   W1T1NG
249612       W1U       1LP   W1U1LP
249613       W1U       1LP   W1U1LP
249614       W1D       7DT   W1D7DT
249615       W1F       0SB   W1F0SB

[249616 rows x 3 columns]
```

```
df = df.drop(columns=['PostCode1', 'Postcode2'])
print(df)
```

```
       Postcode
0        W1D5AF
1         W21HH
2         W26EY
3       WC2H0EE
4        SW72AP
...         ...
249611   W1T1NG
249612   W1U1LP
```

```
249613   W1U1LP
249614   W1D7DT
249615   W1F0SB

[249616 rows x 1 columns]
```

```
LiNEW = pd.concat([Licensing,df],axis=1).drop_duplicates()
LiNEW.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 249616 entries, 0 to 249615
Data columns (total 12 columns):
 #   Column    Non-Null Count   Dtype 
---  ------    --------------   ----- 
 0   REFVAL    249616 non-null  object
 1   ADDRESS   249616 non-null  string
 2   LIUSE     237886 non-null  object
 3   ISSUED    249616 non-null  object
 4   LIPERMIT  248905 non-null  object
 5   LICYCLE   224063 non-null  object
 6   FTYPE     249616 non-null  object
 7   FVALUE    232187 non-null  object
 8   OPENT     223901 non-null  object
 9   CLOST     223906 non-null  object
 10  STREET    249616 non-null  string
 11  Postcode  244305 non-null  string
dtypes: object(9), string(3)
memory usage: 24.8+ MB
```

```
LiNEW.sample(5, random_state=10)
```

```
                 REFVAL                                           
ADDRESS  \
132116  20/00884/LIPDPS  Saw Swee Hock Centre
1 Sheffield Street
London...   
137387  18/13729/LIPDPS                         210 Strand
London
WC2R 1AP   
65543     20/03567/LIPV                 25 Albemarle Street
London
W1S 4HX   
119562    18/00645/LIPN                 9 Fitzmaurice Place
London
W1J 5JD   
90288   18/03030/LIPDPS            Stratton House
5 Stratton Street
London   

        LIUSE      ISSUED LIPERMIT LICYCLE    FTYPE FVALUE 
OPENT  \
132116  RT199  05/03/2020   ENTOTH   00Z56   EXEMPT     NO  09:00:00

137387  PT303  14/12/2018   LATENR   00Z56  ALCOHOL    YES  23:00:00

65543   PT234  13/04/2021   RETALC   07SUN   NUMBER    NaN  12:00:00

119562  RT061  02/04/2018   ENTIND   00ALL   EXEMPT     NO  07:00:00

90288   PT234  04/04/2018   RETALC   00Z14   NUMBER   <250  10:00:00


           CLOST                  STREET Postcode
132116  03:00:00  Street
London
WC2A 2AP  WC2A2AP
137387  00:00:00  Strand
London
WC2R 1AP  WC2R1AP
65543   22:30:00   Street
London
W1S 4HX   W1S4HX
119562  01:00:00    Place
London
W1J 5JD   W1J5JD
90288   23:30:00  Stratton Street
London     <NA>
```

```
LiFINAL = LiNEW.drop(columns=['ADDRESS', 'ISSUED', 'OPENT', 'CLOST',
'STREET'])
LiFINAL.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 249616 entries, 0 to 249615
Data columns (total 7 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   REFVAL    249616 non-null  object
 1   LIUSE     237886 non-null  object
 2   LIPERMIT  248905 non-null  object
 3   LICYCLE   224063 non-null  object
 4   FTYPE     249616 non-null  object
 5   FVALUE    232187 non-null  object
 6   Postcode  244305 non-null  string
dtypes: object(6), string(1)
memory usage: 15.2+ MB
```

```
Postcodes_all = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Profile/postcodes.csv')
Postcodes_all.info()
Postcodes_all.sample(5, random_state=10)
```

```
        Postcode In Use?   Latitude  Longitude   Easting  Northing
Grid Ref  \
1678345  NW3 6TY     Yes  51.556703  -0.179681  526292.0  185772.0
TQ262857
2572440  WF5 9QH     Yes  53.691003  -1.593168  426963.0  421694.0
SE269216
174915   BD9 6AE     Yes  53.807679  -1.805303  412919.0  434616.0
SE129346
1734150  OX5 2HU     Yes  51.820920  -1.278960  449795.0  213834.0
SP497138
899403   G61 3JL     Yes  55.930167  -4.334004  254280.0  673250.0
NS542732

                 County             District             Ward  ...  \
1678345  Greater London               Camden          Frognal  ...
2572440  West Yorkshire             Kirklees    Dewsbury East  ...
174915   West Yorkshire             Bradford           Toller  ...
1734150     Oxfordshire             Cherwell  Kidlington East  ...
899403              NaN  East Dunbartonshire   Bearsden North  ...

                Police force    Water company    Plus Code Average
Income  \
1678345  Metropolitan Police     Thames Water  9C3XHR4C+M4
67300.0
2572440       West Yorkshire  Yorkshire Water  9C5WMCR4+CP
33600.0
174915        West Yorkshire  Yorkshire Water  9C5WR55V+3V
36600.0
1734150        Thames Valley     Thames Water  9C3WRPCC+9C
51400.0
899403              Scotland   Scottish Water  9C7QWMJ8+39
NaN

        Sewage Company  Travel To Work Area  \
1678345            NaN               London
2572440            NaN         Huddersfield
174915             NaN             Bradford
1734150            NaN               Oxford
899403             NaN              Glasgow

                                        ITL level 2  \
1678345                         Inner London - West
2572440                              West Yorkshire
174915                               West Yorkshire
1734150  Berkshire, Buckinghamshire and Oxfordshire
899403                        West Central Scotland

                                               ITL level 3  \
1678345                          Camden and City of London
2572440                            Calderdale and Kirklees
```

```
174915                                            Bradford
1734150                                     Oxfordshire CC
899403   East Dunbartonshire, West Dunbartonshire, and ...

                                                     UPRNs  Distance
to sea
1678345  5000359,5000360,5014722,5014723,5019136,501913...
42.4103
2572440  63173967,83180040,83180041,83180059,83180060,8...
58.7540
174915   10010595783,10023344592,10070057448,1009011697...
70.8299
1734150  10011892799,10011919223,10011923860,1001207898...
96.0237
899403   132046516,132046517,132046518,132046519,132046...
11.2847

[5 rows x 53 columns]
```

```
Postcodes = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Profile/postcodes.csv', usecols =
['Postcode','District', 'Ward', 'District Code', 'Ward Code', 'Census
output area'])
Postcodes.info()
Postcodes.sample(5, random_state=10)
```

```
        Postcode             District             Ward District
Code  \
1678345  NW3 6TY               Camden          Frognal     E09000007

2572440  WF5 9QH             Kirklees    Dewsbury East     E08000034

174915   BD9 6AE             Bradford           Toller     E08000032

1734150  OX5 2HU             Cherwell  Kidlington East     E07000177

899403   G61 3JL  East Dunbartonshire   Bearsden North     S12000045


         Ward Code Census output area
1678345  E05013657          E00004355
2572440  E05001398          E00056078
174915   E05001364          E00054503
1734150  E05010932          E00145173
899403   S13002902          S00100995
```

```
Postcodes_WCC = Postcodes[Postcodes["District"] == 'Westminster']

Postcodes_WCC.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 31085 entries, 774984 to 2541967
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Postcode            31085 non-null  object
 1   District            31085 non-null  object
 2   Ward                31085 non-null  object
 3   District Code       31085 non-null  object
 4   Ward Code           31085 non-null  object
 5   Census output area  31085 non-null  object
dtypes: object(6)
memory usage: 1.7+ MB
```

```
Postcodes_WCC.sample(5, random_state=10)
```

```
        Postcode     District        Ward District Code  Ward Code  \
2482146  W1H 3FN  Westminster  Marylebone     E09000033  E05013801
1687164  NW8 8QH  Westminster  Abbey Road     E09000033  E05013792
2491850  W1V 1LA  Westminster    West End     E09000033  E05013808
1687540  NW8 9SG  Westminster  Abbey Road     E09000033  E05013792
2507605   W9 1NP  Westminster  Maida Vale     E09000033  E05013800

        Census output area
2482146          E00023840
1687164          E00023909
2491850          E00024110
1687540          E00023418
2507605          E00023793
```

```
Postcodes_WCC['Postcode'] = Postcodes_WCC['Postcode'].str.replace(" ","")
```

```
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/pandas/core/frame.py:3612: 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
```

```
Postcodes_WCC = Postcodes_WCC.drop(columns={'District Code', 'Ward Code', 'District', 'Ward'})
```

```
Postcodes_WCC.sample(5, random_state=10)
```

```
        Postcode Census output area
2482146   W1H3FN          E00023840
1687164   NW88QH          E00023909
2491850   W1V1LA          E00024110
```

```
1687540   NW89SG          E00023418
2507605    W91NP          E00023793
```

```
Li_Post = pd.merge(LiFINAL, Postcodes_WCC, on=["Postcode"])
```

```
Li_Post.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 238898 entries, 0 to 238897
Data columns (total 8 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   REFVAL              238898 non-null  object
 1   LIUSE               227789 non-null  object
 2   LIPERMIT            238206 non-null  object
 3   LICYCLE             214124 non-null  object
 4   FTYPE               238898 non-null  object
 5   FVALUE              222626 non-null  object
 6   Postcode            238898 non-null  object
 7   Census output area  238898 non-null  object
dtypes: object(8)
memory usage: 16.4+ MB
```

```
Li_Post.sample(5, random_state=10)
```

```
                 REFVAL  LIUSE LIPERMIT LICYCLE    FTYPE FVALUE
Postcode  \
69441   18/04709/LIPDPS  PT234    ENTML   07SUN  ALCOHOL   YESS
W1B4DG
176575    18/12412/LIPN    NaN   LATENR   00ALL   STRUCT   PERM
W1J0DA
5782      18/01603/LIPV  PT234   RETALC   07SUN   NUMBER   <250
SW1Y4PE
229603  18/01419/LIPCHT  PT234   RETALC    00WD   STRUCT   PERM
SW1X7PQ
55108    18/04937/LIPVM  PT199   ENTDAN   07SUN   NUMBER    NaN
W1S1HU

       Census output area
69441           E00024112
176575          E00023935
5782            E00023935
229603          E00175206
55108           E00175190
```

```
Li_Post = Li_Post.rename(columns={'Census output area': 'OAs'})
```

```
Li_Post = Li_Post.drop(columns={'Postcode'})
```

```
Li_Post.sample(5, random_state=10)
```

```
                 REFVAL  LIUSE LIPERMIT LICYCLE    FTYPE FVALUE
OAs
69441   18/04709/LIPDPS  PT234    ENTML   07SUN  ALCOHOL   YESS
E00024112
176575    18/12412/LIPN    NaN   LATENR   00ALL   STRUCT   PERM
E00023935
5782      18/01603/LIPV  PT234   RETALC   07SUN   NUMBER   <250
E00023935
229603  18/01419/LIPCHT  PT234   RETALC    00WD   STRUCT   PERM
E00175206
55108    18/04937/LIPVM  PT199   ENTDAN   07SUN   NUMBER    NaN
E00175190
```

```
Li_trial = Li_Post.drop(columns={'REFVAL', 'LICYCLE', 'FVALUE',
'FTYPE', 'LIPERMIT'})
Li_trial.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 238898 entries, 0 to 238897
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   LIUSE   227789 non-null  object
 1   OAs     238898 non-null  object
dtypes: object(2)
memory usage: 5.5+ MB
```

```
Li_trial.sample(5, random_state=10)
```

```
        LIUSE        OAs
69441   PT234  E00024112
176575    NaN  E00023935
5782    PT234  E00023935
229603  PT234  E00175206
55108   PT199  E00175190
```

```
Litable = Li_trial
```

```
Litable["GACLGE"] = " "
Litable["GAVESS"] = " "
Litable["LIMSTL"] = " "
Litable["LIPSL"] = " "
Litable["PT011"] = " "
Litable["PT019"] = " "
Litable["PT031"] = " "
Litable["PT049"] = " "
Litable["PT056"] = " "
Litable["PT057"] = " "
Litable["PT060"] = " "
Litable["PT061"] = " "
Litable["PT062"] = " "
```

```
Litable["PT065"] = " "

Litable["PT070"] = " "
Litable["PT074"] = " "
Litable["PT075"] = " "
Litable["PT082"] = " "
Litable["PT086"] = " "
Litable["PT100"] = " "
Litable["PT104"] = " "
Litable["PT106"] = " "
Litable["PT122"] = " "
Litable["PT135"] = " "

Litable["PT137"] = " "
Litable["PT138"] = " "
Litable["PT139"] = " "
Litable["PT140"] = " "
Litable["PT152"] = " "
Litable["PT154"] = " "
Litable["PT155"] = " "
Litable["PT165"] = " "
Litable["PT189"] = " "
Litable["PT195"] = " "

Litable["PT196"] = " "
Litable["PT199"] = " "
Litable["PT203"] = " "
Litable["PT204"] = " "
Litable["PT209"] = " "
Litable["PT225"] = " "
Litable["PT226"] = " "
Litable["PT227"] = " "
Litable["PT232"] = " "
Litable["PT234"] = " "

Litable["PT243"] = " "
Litable["PT249"] = " "
Litable["PT253"] = " "
Litable["PT259"] = " "
Litable["PT260"] = " "
Litable["PT270"] = " "
Litable["PT279"] = " "
Litable["PT284"] = " "
Litable["PT288"] = " "
Litable["PT293"] = " "

Litable["PT303"] = " "
Litable["PT304"] = " "
Litable["PT409"] = " "
Litable["PT417"] = " "
```

```python
Litable["PT437"] = " "
Litable["PT439"] = " "
Litable["PT442"] = " "
Litable["PT500"] = " "
Litable["PT504"] = " "
Litable["PT508"] = " "

Litable["PT993"] = " "
Litable["PT995"] = " "
Litable["PT998"] = " "
Litable["PT999"] = " "
Litable["RT061"] = " "
Litable["RT199"] = " "
Litable["RT442"] = " "
Litable["RT226"] = " "
Litable["RT234"] = " "
Litable["RT303"] = " "
Litable["SEV"] = " "

Litable.sample(5, random_state=10)

Litable = Litable.fillna(0)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'GACLGE' :
        Litable.loc[index, 'GACLGE'] = '1'

    else:
        Litable.loc[index, 'GACLGE'] = '0'

print(Litable)
Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'GAVESS' :
        Litable.loc[index, 'GAVESS'] = '1'

    else:
        Litable.loc[index, 'GAVESS'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'LIMSTL' :
        Litable.loc[index, 'LIMSTL'] = '1'

    else:
        Litable.loc[index, 'LIMSTL'] = '0'

Litable.sample(5, random_state=10)
```

```
for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'LIPSL' :
        Litable.loc[index, 'LIPSL'] = '1'

    else:
        Litable.loc[index, 'LIPSL'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT011' :
        Litable.loc[index, 'PT011'] = '1'

    else:
        Litable.loc[index, 'PT011'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT019' :
        Litable.loc[index, 'PT019'] = '1'

    else:
        Litable.loc[index, 'PT019'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT031' :
        Litable.loc[index, 'PT031'] = '1'

    else:
        Litable.loc[index, 'PT031'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT049' :
        Litable.loc[index, 'PT049'] = '1'

    else:
        Litable.loc[index, 'PT049'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT056' :
        Litable.loc[index, 'PT056'] = '1'

    else:
        Litable.loc[index, 'PT056'] = '0'
Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT057' :
        Litable.loc[index, 'PT057'] = '1'

    else:
        Litable.loc[index, 'PT057'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT060' :
        Litable.loc[index, 'PT060'] = '1'

    else:
        Litable.loc[index, 'PT060'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT061' :
        Litable.loc[index, 'PT061'] = '1'

    else:
        Litable.loc[index, 'PT061'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT062' :
        Litable.loc[index, 'PT062'] = '1'

    else:
        Litable.loc[index, 'PT062'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT065' :
        Litable.loc[index, 'PT065'] = '1'

    else:
        Litable.loc[index, 'PT065'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT070' :
        Litable.loc[index, 'PT070'] = '1'
    else:
        Litable.loc[index, 'PT070'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT074' :
        Litable.loc[index, 'PT074'] = '1'

    else:
        Litable.loc[index, 'PT074'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT075' :
        Litable.loc[index, 'PT075'] = '1'

    else:
        Litable.loc[index, 'PT075'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT082' :
        Litable.loc[index, 'PT082'] = '1'

    else:
        Litable.loc[index, 'PT082'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT086' :
        Litable.loc[index, 'PT086'] = '1'

    else:
        Litable.loc[index, 'PT086'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT100' :
        Litable.loc[index, 'PT100'] = '1'

    else:
        Litable.loc[index, 'PT100'] = '0'

Litable.sample(5, random_state=10)
```

```
for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT104' :
        Litable.loc[index, 'PT104'] = '1'

    else:
        Litable.loc[index, 'PT104'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT106' :
        Litable.loc[index, 'PT106'] = '1'

    else:
        Litable.loc[index, 'PT106'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT122' :
        Litable.loc[index, 'PT122'] = '1'

    else:
        Litable.loc[index, 'PT122'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT135' :
        Litable.loc[index, 'PT135'] = '1'

    else:
        Litable.loc[index, 'PT135'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT137' :
        Litable.loc[index, 'PT137'] = '1'

    else:
        Litable.loc[index, 'PT137'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT138' :
        Litable.loc[index, 'PT138'] = '1'

    else:
        Litable.loc[index, 'PT138'] = '0'
Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT139' :
        Litable.loc[index, 'PT139'] = '1'

    else:
        Litable.loc[index, 'PT139'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT140' :
        Litable.loc[index, 'PT140'] = '1'

    else:
        Litable.loc[index, 'PT140'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT152' :
        Litable.loc[index, 'PT152'] = '1'

    else:
        Litable.loc[index, 'PT152'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT154' :
        Litable.loc[index, 'PT154'] = '1'

    else:
        Litable.loc[index, 'PT154'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT155' :
        Litable.loc[index, 'PT155'] = '1'

    else:
        Litable.loc[index, 'PT155'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT165' :
        Litable.loc[index, 'PT165'] = '1'
    else:
        Litable.loc[index, 'PT165'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT189' :
        Litable.loc[index, 'PT189'] = '1'

    else:
        Litable.loc[index, 'PT189'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT195' :
        Litable.loc[index, 'PT195'] = '1'

    else:
        Litable.loc[index, 'PT195'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT196' :
        Litable.loc[index, 'PT196'] = '1'

    else:
        Litable.loc[index, 'PT196'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT199' :
        Litable.loc[index, 'PT199'] = '1'

    else:
        Litable.loc[index, 'PT199'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT203' :
        Litable.loc[index, 'PT203'] = '1'

    else:
        Litable.loc[index, 'PT203'] = '0'

Litable.sample(5, random_state=10)
```

```
for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT204' :
        Litable.loc[index, 'PT204'] = '1'

    else:
        Litable.loc[index, 'PT204'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT209' :
        Litable.loc[index, 'PT209'] = '1'

    else:
        Litable.loc[index, 'PT209'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT225' :
        Litable.loc[index, 'PT225'] = '1'

    else:
        Litable.loc[index, 'PT225'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT226' :
        Litable.loc[index, 'PT226'] = '1'

    else:
        Litable.loc[index, 'PT226'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT227' :
        Litable.loc[index, 'PT227'] = '1'

    else:
        Litable.loc[index, 'PT227'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT232' :
        Litable.loc[index, 'PT232'] = '1'

    else:
        Litable.loc[index, 'PT232'] = '0'
Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT234' :
        Litable.loc[index, 'PT234'] = '1'

    else:
        Litable.loc[index, 'PT234'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT243' :
        Litable.loc[index, 'PT243'] = '1'

    else:
        Litable.loc[index, 'PT243'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT249' :
        Litable.loc[index, 'PT249'] = '1'

    else:
        Litable.loc[index, 'PT249'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT253' :
        Litable.loc[index, 'PT253'] = '1'

    else:
        Litable.loc[index, 'PT253'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT259' :
        Litable.loc[index, 'PT259'] = '1'

    else:
        Litable.loc[index, 'PT259'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT260' :
        Litable.loc[index, 'PT260'] = '1'
    else:
        Litable.loc[index, 'PT260'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT270' :
        Litable.loc[index, 'PT270'] = '1'

    else:
        Litable.loc[index, 'PT270'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT279' :
        Litable.loc[index, 'PT279'] = '1'

    else:
        Litable.loc[index, 'PT279'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT284' :
        Litable.loc[index, 'PT284'] = '1'

    else:
        Litable.loc[index, 'PT284'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT288' :
        Litable.loc[index, 'PT288'] = '1'

    else:
        Litable.loc[index, 'PT288'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT293' :
        Litable.loc[index, 'PT293'] = '1'

    else:
        Litable.loc[index, 'PT293'] = '0'

Litable.sample(5, random_state=10)
```

```
for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT303' :
        Litable.loc[index, 'PT303'] = '1'

    else:
        Litable.loc[index, 'PT303'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT304' :
        Litable.loc[index, 'PT304'] = '1'

    else:
        Litable.loc[index, 'PT304'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT303' :
        Litable.loc[index, 'PT303'] = '1'

    else:
        Litable.loc[index, 'PT303'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT304' :
        Litable.loc[index, 'PT304'] = '1'

    else:
        Litable.loc[index, 'PT304'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT409' :
        Litable.loc[index, 'PT409'] = '1'

    else:
        Litable.loc[index, 'PT409'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT417' :
        Litable.loc[index, 'PT417'] = '1'

    else:
        Litable.loc[index, 'PT417'] = '0'
Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT437' :
        Litable.loc[index, 'PT437'] = '1'

    else:
        Litable.loc[index, 'PT437'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT439' :
        Litable.loc[index, 'PT439'] = '1'

    else:
        Litable.loc[index, 'PT439'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT437' :
        Litable.loc[index, 'PT437'] = '1'

    else:
        Litable.loc[index, 'PT437'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT439' :
        Litable.loc[index, 'PT439'] = '1'

    else:
        Litable.loc[index, 'PT439'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT442' :
        Litable.loc[index, 'PT442'] = '1'

    else:
        Litable.loc[index, 'PT442'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT500' :
        Litable.loc[index, 'PT500'] = '1'
    else:
        Litable.loc[index, 'PT500'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT504' :
        Litable.loc[index, 'PT504'] = '1'

    else:
        Litable.loc[index, 'PT504'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT508' :
        Litable.loc[index, 'PT508'] = '1'

    else:
        Litable.loc[index, 'PT508'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT993' :
        Litable.loc[index, 'PT993'] = '1'

    else:
        Litable.loc[index, 'PT993'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT995' :
        Litable.loc[index, 'PT995'] = '1'

    else:
        Litable.loc[index, 'PT995'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT998' :
        Litable.loc[index, 'PT998'] = '1'

    else:
        Litable.loc[index, 'PT998'] = '0'

Litable.sample(5, random_state=10)
```

```
for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'PT999' :
        Litable.loc[index, 'PT999'] = '1'

    else:
        Litable.loc[index, 'PT999'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'RT061' :
        Litable.loc[index, 'RT061'] = '1'

    else:
        Litable.loc[index, 'RT061'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'RT199' :
        Litable.loc[index, 'RT199'] = '1'

    else:
        Litable.loc[index, 'RT199'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'RT442' :
        Litable.loc[index, 'RT442'] = '1'

    else:
        Litable.loc[index, 'RT442'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'RT226' :
        Litable.loc[index, 'RT226'] = '1'

    else:
        Litable.loc[index, 'RT226'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'RT234' :
        Litable.loc[index, 'RT234'] = '1'

    else:
        Litable.loc[index, 'RT234'] = '0'
Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'RT303' :
        Litable.loc[index, 'RT303'] = '1'

    else:
        Litable.loc[index, 'RT303'] = '0'

Litable.sample(5, random_state=10)

for index, row in Litable.iterrows() :
    if Litable.loc[index, 'LIUSE'] == 'SEV' :
        Litable.loc[index, 'SEV'] = '1'

    else:
        Litable.loc[index, 'SEV'] = '0'

Litable.sample(5, random_state=10)

Litable = Litable.drop(columns={'LIUSE'})

Litable.sample(5, random_state=10)

Litable["GACLGE"] = Litable["GACLGE"].astype(float)
Litable["GAVESS"] = Litable["GAVESS"].astype(float)
Litable["LIMSTL"] = Litable["LIMSTL"].astype(float)
Litable["LIPSL"] = Litable["LIPSL"].astype(float)
Litable["PT011"] = Litable["PT011"].astype(float)
Litable["PT019"] = Litable["PT019"].astype(float)
Litable["PT031"] = Litable["PT031"].astype(float)
Litable["PT049"] = Litable["PT049"].astype(float)
Litable["PT056"] = Litable["PT056"].astype(float)
Litable["PT057"] = Litable["PT057"].astype(float)
Litable["PT060"] = Litable["PT060"].astype(float)
Litable["PT061"] = Litable["PT061"].astype(float)
Litable["PT062"] = Litable["PT062"].astype(float)
Litable["PT065"] = Litable["PT065"].astype(float)


Litable["PT070"] = Litable["PT070"].astype(float)
Litable["PT074"] = Litable["PT074"].astype(float)
Litable["PT075"] = Litable["PT075"].astype(float)
Litable["PT082"] = Litable["PT082"].astype(float)
Litable["PT086"] = Litable["PT086"].astype(float)
Litable["PT100"] = Litable["PT100"].astype(float)
Litable["PT104"] = Litable["PT104"].astype(float)
Litable["PT106"] = Litable["PT106"].astype(float)
Litable["PT122"] = Litable["PT122"].astype(float)
Litable["PT135"] = Litable["PT135"].astype(float)
```

```
Litable["PT137"] = Litable["PT137"].astype(float)
Litable["PT138"] = Litable["PT138"].astype(float)
Litable["PT139"] = Litable["PT139"].astype(float)
Litable["PT140"] = Litable["PT140"].astype(float)
Litable["PT152"] = Litable["PT152"].astype(float)
Litable["PT154"] = Litable["PT154"].astype(float)
Litable["PT155"] = Litable["PT155"].astype(float)
Litable["PT165"] = Litable["PT165"].astype(float)
Litable["PT189"] = Litable["PT189"].astype(float)
Litable["PT195"] = Litable["PT195"].astype(float)

Litable["PT196"] = Litable["PT196"].astype(float)
Litable["PT199"] = Litable["PT199"].astype(float)
Litable["PT203"] = Litable["PT203"].astype(float)
Litable["PT204"] = Litable["PT204"].astype(float)
Litable["PT209"] = Litable["PT209"].astype(float)
Litable["PT225"] = Litable["PT225"].astype(float)
Litable["PT226"] = Litable["PT226"].astype(float)
Litable["PT227"] = Litable["PT227"].astype(float)
Litable["PT232"] = Litable["PT232"].astype(float)
Litable["PT234"] = Litable["PT234"].astype(float)

Litable["PT243"] = Litable["PT243"].astype(float)
Litable["PT249"] = Litable["PT249"].astype(float)
Litable["PT253"] = Litable["PT253"].astype(float)
Litable["PT259"] = Litable["PT259"].astype(float)
Litable["PT260"] = Litable["PT260"].astype(float)
Litable["PT270"] = Litable["PT270"].astype(float)
Litable["PT279"] = Litable["PT279"].astype(float)
Litable["PT284"] = Litable["PT284"].astype(float)
Litable["PT288"] = Litable["PT288"].astype(float)
Litable["PT293"] = Litable["PT293"].astype(float)

Litable["PT303"] = Litable["PT303"].astype(float)
Litable["PT304"] = Litable["PT304"].astype(float)
Litable["PT409"] = Litable["PT409"].astype(float)
Litable["PT417"] = Litable["PT417"].astype(float)
Litable["PT437"] = Litable["PT437"].astype(float)
Litable["PT439"] = Litable["PT439"].astype(float)
Litable["PT442"] = Litable["PT442"].astype(float)
Litable["PT500"] = Litable["PT500"].astype(float)
Litable["PT504"] = Litable["PT504"].astype(float)
Litable["PT508"] = Litable["PT508"].astype(float)

Litable["PT993"] = Litable["PT993"].astype(float)
Litable["PT995"] = Litable["PT995"].astype(float)
Litable["PT998"] = Litable["PT998"].astype(float)
Litable["PT999"] = Litable["PT999"].astype(float)
Litable["RT061"] = Litable["RT061"].astype(float)
Litable["RT199"] = Litable["RT199"].astype(float)
```

```
Litable["RT442"] = Litable["RT442"].astype(float)
Litable["RT226"] = Litable["RT226"].astype(float)
Litable["RT234"] = Litable["RT234"].astype(float)
Litable["RT303"] = Litable["RT303"].astype(float)
Litable["SEV"] = Litable["SEV"].astype(float)

Litable.info()

Litable.to_csv('Licensing.csv', encoding='utf-8', index=False)

Licensing = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/Final_cleanData/Licensing.csv')
Licensing.info()

Licensing ["Licensing_all"] = " "

Licensing ["Licensing_all"] = Licensing.sum(axis=1)

Licensing.sample(15)

Licensing.to_csv('Licensing.csv', encoding='utf-8', index=False)
```

Final datasets

```
# WCC paycheck and households
PayCheck = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/Final_cleanData/PayCheck.csv')
PayCheck.info()

<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 783 entries, 0 to 782
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   OAs               783 non-null    object 
 1   Total households  783 non-null    int64  
 2   Mean Income       779 non-null    float64
 3   Median Income     779 non-null    float64
 4   Mode Income       779 non-null    float64
 5   Lower Quartile    779 non-null    float64
dtypes: float64(4), int64(1), object(1)
memory usage: 36.8+ KB

PayCheck.sample(5, random_state=10)

           OAs  Total households   Mean Income  Median Income  Mode
Income  \
27   E00023437               110  74502.727273   67158.203125
110000.0
305  E00023732               131  70203.816794   63118.361153
110000.0
579  E00024023               184  56000.543478   48666.069830
42500.0
```

```
514  E00023949              136  47063.235294   40568.295115
27500.0
181  E00023599              127  46676.377953   40325.027086
27500.0

     Lower Quartile
27     43004.966887
305    40721.649485
579    29600.790514
514    24663.716814
181    24219.790676
```

```
# WCC ambulance call outs
Ambulance = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/Final_cleanData/Ambulance.csv')
Ambulance.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 56711772 entries, 0 to 56711771
Data columns (total 2 columns):
 #   Column    Dtype 
---  ------    ----- 
 0   OAs       object
 1   LonAmALL  int64 
dtypes: int64(1), object(1)
memory usage: 865.4+ MB
```

```
Ambulance = Ambulance.groupby(by=['OAs'],
dropna=False).sum().reset_index()
Ambulance.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 794 entries, 0 to 793
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   OAs       794 non-null    object
 1   LonAmALL  794 non-null    int64 
dtypes: int64(1), object(1)
memory usage: 12.5+ KB
```

```
Ambulance.sample(5, random_state=10)
```

```
           OAs  LonAmALL
124  E00023528      2000
386  E00023810      2132
163  E00023570     67200
52   E00023455      6426
27   E00023430      1102
```

```
# WCC crime data compiled
Crime = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/Final_cleanData/Crime.csv')
Crime.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 209196 entries, 0 to 209195
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   OAs                         204572 non-null  object 
 1   Damage_incident             14077 non-null   float64
 2   Burglary_incident           19994 non-null   float64
 3   Disorder_incident           78977 non-null   float64
 4   Fraud_incident              1960 non-null    float64
 5   Robbery_incident            16828 non-null   float64
 6   SexRelated_incident         3548 non-null    float64
 7   Violence_incident           71614 non-null   float64
 8   WeaponPossession_incident   2198 non-null    float64
 9   Crime_all                   209196 non-null  float64
dtypes: float64(9), object(1)
memory usage: 16.0+ MB
```

```
Crime = Crime.groupby(by=['OAs'], dropna=False).sum().reset_index()
Crime.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 747 entries, 0 to 746
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   OAs                         746 non-null    object 
 1   Damage_incident             747 non-null    float64
 2   Burglary_incident           747 non-null    float64
 3   Disorder_incident           747 non-null    float64
 4   Fraud_incident              747 non-null    float64
 5   Robbery_incident            747 non-null    float64
 6   SexRelated_incident         747 non-null    float64
 7   Violence_incident           747 non-null    float64
 8   WeaponPossession_incident   747 non-null    float64
 9   Crime_all                   747 non-null    float64
dtypes: float64(9), object(1)
memory usage: 58.5+ KB
```

```
Crime.sample(5, random_state=10)
```

```
           OAs  Damage_incident  Burglary_incident  Disorder_incident 
\
240  E00023662             15.0               14.0               27.0 

220  E00023639              2.0               10.0               27.0 

222  E00023641              5.0               25.0               17.0 
```

```
372  E00023814             11.0                 11.0                 6.0

341  E00023777              2.0                 20.0                13.0


     Fraud_incident  Robbery_incident  SexRelated_incident
Violence_incident  \
240             0.0               7.0                  0.0
43.0
220             5.0               3.0                  0.0
14.0
222             0.0               3.0                  0.0
40.0
372             0.0               0.0                  0.0
29.0
341             0.0               2.0                  0.0
9.0

     WeaponPossession_incident  Crime_all
240                        0.0      106.0
220                        0.0       61.0
222                        0.0       90.0
372                        0.0       57.0
341                        0.0       46.0
```

```
# WCC noise data compiled
Noise_all = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/Final_cleanData/Noise.csv')
Noise_all.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 90798 entries, 0 to 90797
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   OAs                        90798 non-null  object 
 1   Noise_IncidentALL          90798 non-null  int64  
 2   Animal                     90798 non-null  float64
 3   Building Site              90798 non-null  float64
 4   Commercial Premises        90798 non-null  float64
 5   Email Complaint (1d)       90798 non-null  float64
 6   Formal complaints          90798 non-null  float64
 7   Non Noise Complaint (45m)  90798 non-null  float64
 8   Non Noise Complaint (4d)   90798 non-null  float64
 9   Proactive Noise            90798 non-null  float64
 10  Property Alarm             90798 non-null  float64
 11  Residential Premises       90798 non-null  float64
 12  Street                     90798 non-null  float64
 13  VIP complaint              90798 non-null  float64
```

```
dtypes: float64(12), int64(1), object(1)
memory usage: 9.7+ MB
```

```
Noise_all = Noise_all.groupby(by=['OAs'], dropna=True,
as_index=False).sum()
Noise_all.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 839 entries, 0 to 838
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   OAs                        839 non-null    object 
 1   Noise_IncidentALL          839 non-null    int64  
 2   Animal                     839 non-null    float64
 3   Building Site              839 non-null    float64
 4   Commercial Premises        839 non-null    float64
 5   Email Complaint (1d)       839 non-null    float64
 6   Formal complaints          839 non-null    float64
 7   Non Noise Complaint (45m)  839 non-null    float64
 8   Non Noise Complaint (4d)   839 non-null    float64
 9   Proactive Noise            839 non-null    float64
 10  Property Alarm             839 non-null    float64
 11  Residential Premises       839 non-null    float64
 12  Street                     839 non-null    float64
 13  VIP complaint              839 non-null    float64
dtypes: float64(12), int64(1), object(1)
memory usage: 91.9+ KB
```

```
Noise_all.sample(10)
```

```
           OAs  Noise_IncidentALL  Animal  Building Site  Commercial
Premises  \
229  E00023596                145     3.0            2.0
0.0
628  E00024023                 60     0.0            2.0
0.0
805  E00175245                 87     1.0            6.0
0.0
354  E00023730                 24     0.0            3.0
1.0
585  E00023974                 31     0.0            3.0
0.0
546  E00023930                762     2.0           69.0
147.0
528  E00023911                 27     0.0            0.0
0.0
613  E00024007                 26     0.0            7.0
2.0
283  E00023653                178     8.0            3.0
1.0
```

```
43   E00014315                  4     0.0            0.0
0.0

     Email Complaint (1d)  Formal complaints  Non Noise Complaint
(45m)  \
229                   0.0                0.0
2.0
628                   0.0                0.0
0.0
805                   0.0                0.0
0.0
354                   0.0                0.0
1.0
585                   0.0                0.0
0.0
546                   0.0                0.0
13.0
528                   0.0                0.0
0.0
613                   0.0                0.0
3.0
283                   0.0                0.0
5.0
43                    0.0                0.0
0.0

     Non Noise Complaint (4d)  Proactive Noise  Property Alarm  \
229                       1.0              1.0             1.0
628                       0.0              0.0             4.0
805                       1.0              0.0             3.0
354                       0.0              0.0             0.0
585                       0.0              0.0             0.0
546                       8.0              0.0            44.0
528                       0.0              0.0             0.0
613                       1.0              0.0             2.0
283                       1.0              0.0             3.0
43                        0.0              0.0             0.0

     Residential Premises  Street  VIP complaint
229                 112.0    23.0            0.0
628                  45.0     9.0            0.0
805                  64.0    12.0            0.0
354                  18.0     1.0            0.0
585                  28.0     0.0            0.0
546                 181.0   298.0            0.0
528                  25.0     2.0            0.0
613                   7.0     4.0            0.0
283                 107.0    50.0            0.0
43                    0.0     4.0            0.0
```

```
# Licensing data
Licensing = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/Final_cleanData/Licensing.csv')
Licensing.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 238898 entries, 0 to 238897
Data columns (total 77 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   OAs             238898 non-null  object
 1   GACLGE          238898 non-null  float64
 2   GAVESS          238898 non-null  float64
 3   LIMSTL          238898 non-null  float64
 4   LIPSL           238898 non-null  float64
 5   PT011           238898 non-null  float64
 6   PT019           238898 non-null  float64
 7   PT031           238898 non-null  float64
 8   PT049           238898 non-null  float64
 9   PT056           238898 non-null  float64
 10  PT057           238898 non-null  float64
 11  PT060           238898 non-null  float64
 12  PT061           238898 non-null  float64
 13  PT062           238898 non-null  float64
 14  PT065           238898 non-null  float64
 15  PT070           238898 non-null  float64
 16  PT074           238898 non-null  float64
 17  PT075           238898 non-null  float64
 18  PT082           238898 non-null  float64
 19  PT086           238898 non-null  float64
 20  PT100           238898 non-null  float64
 21  PT104           238898 non-null  float64
 22  PT106           238898 non-null  float64
 23  PT122           238898 non-null  float64
 24  PT135           238898 non-null  float64
 25  PT137           238898 non-null  float64
 26  PT138           238898 non-null  float64
 27  PT139           238898 non-null  float64
 28  PT140           238898 non-null  float64
 29  PT152           238898 non-null  float64
 30  PT154           238898 non-null  float64
 31  PT155           238898 non-null  float64
 32  PT165           238898 non-null  float64
 33  PT189           238898 non-null  float64
 34  PT195           238898 non-null  float64
 35  PT196           238898 non-null  float64
 36  PT199           238898 non-null  float64
 37  PT203           238898 non-null  float64
 38  PT204           238898 non-null  float64
 39  PT209           238898 non-null  float64
 40  PT225          238898 non-null  float64
 41  PT226          238898 non-null  float64
 42  PT227          238898 non-null  float64
 43  PT232          238898 non-null  float64
 44  PT234          238898 non-null  float64
 45  PT243          238898 non-null  float64
 46  PT249          238898 non-null  float64
 47  PT253          238898 non-null  float64
 48  PT259          238898 non-null  float64
 49  PT260          238898 non-null  float64
 50  PT270          238898 non-null  float64
 51  PT279          238898 non-null  float64
 52  PT284          238898 non-null  float64
 53  PT288          238898 non-null  float64
 54  PT293          238898 non-null  float64
 55  PT303          238898 non-null  float64
 56  PT304          238898 non-null  float64
 57  PT409          238898 non-null  float64
 58  PT417          238898 non-null  float64
 59  PT437          238898 non-null  float64
 60  PT439          238898 non-null  float64
 61  PT442          238898 non-null  float64
 62  PT500          238898 non-null  float64
 63  PT504          238898 non-null  float64
 64  PT508          238898 non-null  float64
 65  PT993          238898 non-null  float64
 66  PT995          238898 non-null  float64
 67  PT998          238898 non-null  float64
 68  PT999          238898 non-null  float64
 69  RT061          238898 non-null  float64
 70  RT199          238898 non-null  float64
 71  RT442          238898 non-null  float64
 72  RT226          238898 non-null  float64
 73  RT234          238898 non-null  float64
 74  RT303          238898 non-null  float64
 75  SEV            238898 non-null  float64
 76  Licensing_all  238898 non-null  float64
dtypes: float64(76), object(1)
memory usage: 140.3+ MB
```

```
Licensing = Licensing.groupby(by=['OAs'],
dropna=False).sum().reset_index()
Licensing.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 328 entries, 0 to 327
Data columns (total 77 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   OAs             328 non-null    object
 1   GACLGE          328 non-null    float64
```

```
2   GAVESS         328 non-null    float64
3   LIMSTL         328 non-null    float64
4   LIPSL          328 non-null    float64
5   PT011          328 non-null    float64
6   PT019          328 non-null    float64
7   PT031          328 non-null    float64
8   PT049          328 non-null    float64
9   PT056          328 non-null    float64
10  PT057          328 non-null    float64
11  PT060          328 non-null    float64
12  PT061          328 non-null    float64
13  PT062          328 non-null    float64
14  PT065          328 non-null    float64
15  PT070          328 non-null    float64
16  PT074          328 non-null    float64
17  PT075          328 non-null    float64
18  PT082          328 non-null    float64
19  PT086          328 non-null    float64
20  PT100          328 non-null    float64
21  PT104          328 non-null    float64
22  PT106          328 non-null    float64
23  PT122          328 non-null    float64
24  PT135          328 non-null    float64
25  PT137          328 non-null    float64
26  PT138          328 non-null    float64
27  PT139          328 non-null    float64
28  PT140          328 non-null    float64
29  PT152          328 non-null    float64
30  PT154          328 non-null    float64
31  PT155          328 non-null    float64
32  PT165          328 non-null    float64
33  PT189          328 non-null    float64
34  PT195          328 non-null    float64
35  PT196          328 non-null    float64
36  PT199          328 non-null    float64
37  PT203          328 non-null    float64
38  PT204          328 non-null    float64
39  PT209          328 non-null    float64
40  PT225          328 non-null    float64
41  PT226          328 non-null    float64
42  PT227          328 non-null    float64
43  PT232          328 non-null    float64
44  PT234          328 non-null    float64
45  PT243          328 non-null    float64
46  PT249          328 non-null    float64
47  PT253          328 non-null    float64
48  PT259          328 non-null    float64
49  PT260          328 non-null    float64
50  PT270          328 non-null    float64
51  PT279          328 non-null    float64
```

```
 52  PT284          328 non-null    float64
 53  PT288          328 non-null    float64
 54  PT293          328 non-null    float64
 55  PT303          328 non-null    float64
 56  PT304          328 non-null    float64
 57  PT409          328 non-null    float64
 58  PT417          328 non-null    float64
 59  PT437          328 non-null    float64
 60  PT439          328 non-null    float64
 61  PT442          328 non-null    float64
 62  PT500          328 non-null    float64
 63  PT504          328 non-null    float64
 64  PT508          328 non-null    float64
 65  PT993          328 non-null    float64
 66  PT995          328 non-null    float64
 67  PT998          328 non-null    float64
 68  PT999          328 non-null    float64
 69  RT061          328 non-null    float64
 70  RT199          328 non-null    float64
 71  RT442          328 non-null    float64
 72  RT226          328 non-null    float64
 73  RT234          328 non-null    float64
 74  RT303          328 non-null    float64
 75  SEV            328 non-null    float64
 76  Licensing_all  328 non-null    float64
dtypes: float64(76), object(1)
memory usage: 197.4+ KB
```

```
Licensing.sample(5, random_state=10)
```

```
           OAs  GACLGE  GAVESS  LIMSTL  LIPSL  PT011  PT019  PT031
PT049  \
148  E00023820     0.0     0.0     0.0    0.0    0.0    0.0    0.0
0.0
147  E00023819     0.0     0.0     0.0    0.0    0.0    0.0    0.0
0.0
154  E00023827     0.0     0.0     0.0    0.0    0.0    0.0    0.0
0.0
324  E00175257     0.0     0.0     0.0    0.0    0.0    0.0    0.0
0.0
105  E00023692     0.0     0.0     0.0    0.0    0.0    0.0    0.0
0.0

     PT056  ...  PT998  PT999  RT061  RT199  RT442  RT226  RT234
RT303  SEV  \
148    0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0    0.0
0.0  0.0
147    0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0    0.0
0.0  0.0
154    0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0    0.0
```

```
0.0  0.0
324    0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0    0.0
0.0  0.0
105    0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0    0.0
0.0  0.0

     Licensing_all
148           20.0
147           50.0
154           60.0
324            0.0
105           20.0

[5 rows x 77 columns]
```

Merging these 4 datasets

```
final1 = PayCheck.append(Ambulance)
final1.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 1577 entries, 0 to 793
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   OAs               1577 non-null   object
 1   Total households  783 non-null    float64
 2   Mean Income       779 non-null    float64
 3   Median Income     779 non-null    float64
 4   Mode Income       779 non-null    float64
 5   Lower Quartile    779 non-null    float64
 6   LonAmALL          794 non-null    float64
dtypes: float64(6), object(1)
memory usage: 98.6+ KB
```

```
final2 = final1.append(Crime)
final2.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 2324 entries, 0 to 746
Data columns (total 16 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   OAs                        2323 non-null   object
 1   Total households           783 non-null    float64
 2   Mean Income                779 non-null    float64
 3   Median Income              779 non-null    float64
 4   Mode Income                779 non-null    float64
 5   Lower Quartile             779 non-null    float64
 6   LonAmALL                   794 non-null    float64
 7   Damage_incident            747 non-null    float64
 8   Burglary_incident          747 non-null    float64
 9   Disorder_incident          747 non-null    float64
 10  Fraud_incident             747 non-null    float64
 11  Robbery_incident           747 non-null    float64
 12  SexRelated_incident        747 non-null    float64
 13  Violence_incident          747 non-null    float64
 14  WeaponPossession_incident  747 non-null    float64
 15  Crime_all                  747 non-null    float64
dtypes: float64(15), object(1)
memory usage: 308.7+ KB
```

```
final3 = final2.append(Noise_all)
final3.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 3163 entries, 0 to 838
Data columns (total 29 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   OAs                        3162 non-null   object
 1   Total households           783 non-null    float64
 2   Mean Income                779 non-null    float64
 3   Median Income              779 non-null    float64
 4   Mode Income                779 non-null    float64
 5   Lower Quartile             779 non-null    float64
 6   LonAmALL                   794 non-null    float64
 7   Damage_incident            747 non-null    float64
 8   Burglary_incident          747 non-null    float64
 9   Disorder_incident          747 non-null    float64
 10  Fraud_incident             747 non-null    float64
 11  Robbery_incident           747 non-null    float64
 12  SexRelated_incident        747 non-null    float64
 13  Violence_incident          747 non-null    float64
 14  WeaponPossession_incident  747 non-null    float64
 15  Crime_all                  747 non-null    float64
 16  Noise_IncidentALL          839 non-null    float64
 17  Animal                     839 non-null    float64
 18  Building Site              839 non-null    float64
 19  Commercial Premises        839 non-null    float64
 20  Email Complaint (1d)       839 non-null    float64
 21  Formal complaints          839 non-null    float64
 22  Non Noise Complaint (45m)  839 non-null    float64
 23  Non Noise Complaint (4d)   839 non-null    float64
 24  Proactive Noise            839 non-null    float64
 25  Property Alarm             839 non-null    float64
 26  Residential Premises       839 non-null    float64
 27  Street                     839 non-null    float64
 28  VIP complaint              839 non-null    float64
dtypes: float64(28), object(1)
memory usage: 741.3+ KB
```

```
final4 = final3.append(Licensing)
final4.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
Int64Index: 3491 entries, 0 to 327
Columns: 105 entries, OAs to Licensing_all
dtypes: float64(104), object(1)
memory usage: 2.8+ MB
```

```
CIA=final4.groupby(by=['OAs'], dropna=True).sum().reset_index()
CIA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 850 entries, 0 to 849
Columns: 105 entries, OAs to Licensing_all
dtypes: float64(104), object(1)
memory usage: 697.4+ KB
```

```
CIA ["CIA_Composite"] = " "
```

```
CIA ["CIA_Composite"] = CIA.sum(axis=1)
```

```
/var/folders/c3/qc9fdqbj0lvdcbr2l68p4t2h0000gn/T/
ipykernel_1814/2849925420.py:1: FutureWarning:Dropping of nuisance
columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError.  Select
only valid columns before calling the reduction.
```

```
CIA.sample(10)
```

```
           OAs  Total households   Mean Income  Median Income  Mode
Income  \
651  E00024038             120.0  77214.166667   70674.603175
110000.0
800  E00175230             295.0  40922.711864   34412.388630
17500.0
406  E00023778             104.0  62376.923077   54893.292683
110000.0
566  E00023942              82.0  72647.560976   65883.838384
110000.0
641  E00024026             177.0  53931.638418   46924.119241
27500.0
517  E00023891             159.0  90174.150943   84238.505747
110000.0
564  E00023939             176.0  69548.863636   61893.939394
110000.0
515  E00023889             152.0  51876.315789   44606.946984
27500.0
747  E00174651               0.0      0.000000       0.000000
0.0
385  E00023757              77.0  37810.389610   29875.583204
12500.0
```

```
     Lower Quartile  LonAmALL  Damage_incident  Burglary_incident  \
651    45278.246206    3082.0              0.0                0.0   
800    20580.667594    1364.0              0.0                0.0   
406    34941.860465    7416.0              0.0                0.0   
566    42752.688172  129794.0             20.0                9.0   
641    28878.105590    8960.0              7.0                8.0   
517    54719.562244   50848.0             15.0               21.0   
564    39296.375267  325892.0            144.0              147.0   
515    27290.598291    8840.0              5.0                8.0   
747        0.000000       0.0              0.0                0.0   
385    16637.323944   30704.0             47.0               28.0   

     Disorder_incident  ...  PT999  RT061  RT199  RT442  RT226  RT234  
RT303  \
651                0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
800                0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
406                0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
566               51.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
641               23.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
517               34.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
564              310.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
515                3.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
747                0.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   
385              347.0  ...    0.0    0.0    0.0    0.0    0.0    0.0  
0.0   

     SEV  Licensing_all  CIA_Composite  
651  0.0           30.0  306505.016047  
800  0.0           10.0  115182.768088  
406  0.0            0.0  269844.076225  
566  0.0         1540.0  425012.087531  
641  0.0            0.0  166770.863249  
517  0.0            0.0  390491.218934  
564  0.0         1403.0  612059.178297  
515  0.0          190.0  160977.861064  
747  0.0            0.0     108.000000  
385  0.0           75.0  129044.296758  

[10 rows x 106 columns]
```

```
CIA.to_csv('CIA.csv', encoding='utf-8', index=False)
```

Exploratory Data Analysis or EDA

```
# CIA data called directly
CIA = pd.read_csv('/Users/elika_sinha/Documents/UCL/11.
Dissertation/Term3/Datasets/Final_cleanData/CIA.csv')
CIA.info()
```

```
<class 'lux.core.frame.LuxDataFrame'>
RangeIndex: 850 entries, 0 to 849
Columns: 106 entries, OAs to CIA_Composite
dtypes: float64(105), object(1)
memory usage: 704.0+ KB
```

```
CIA_EDA = ProfileReport(CIA)
```

```
CIA_EDA
```

{"version_major":2,"version_minor":0,"model_id":"ba42b1cc021340f9a248f
58fab5448ef"}

```
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
/Users/elika_sinha/opt/anaconda3/lib/python3.9/site-packages/scipy/
__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is
required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
```

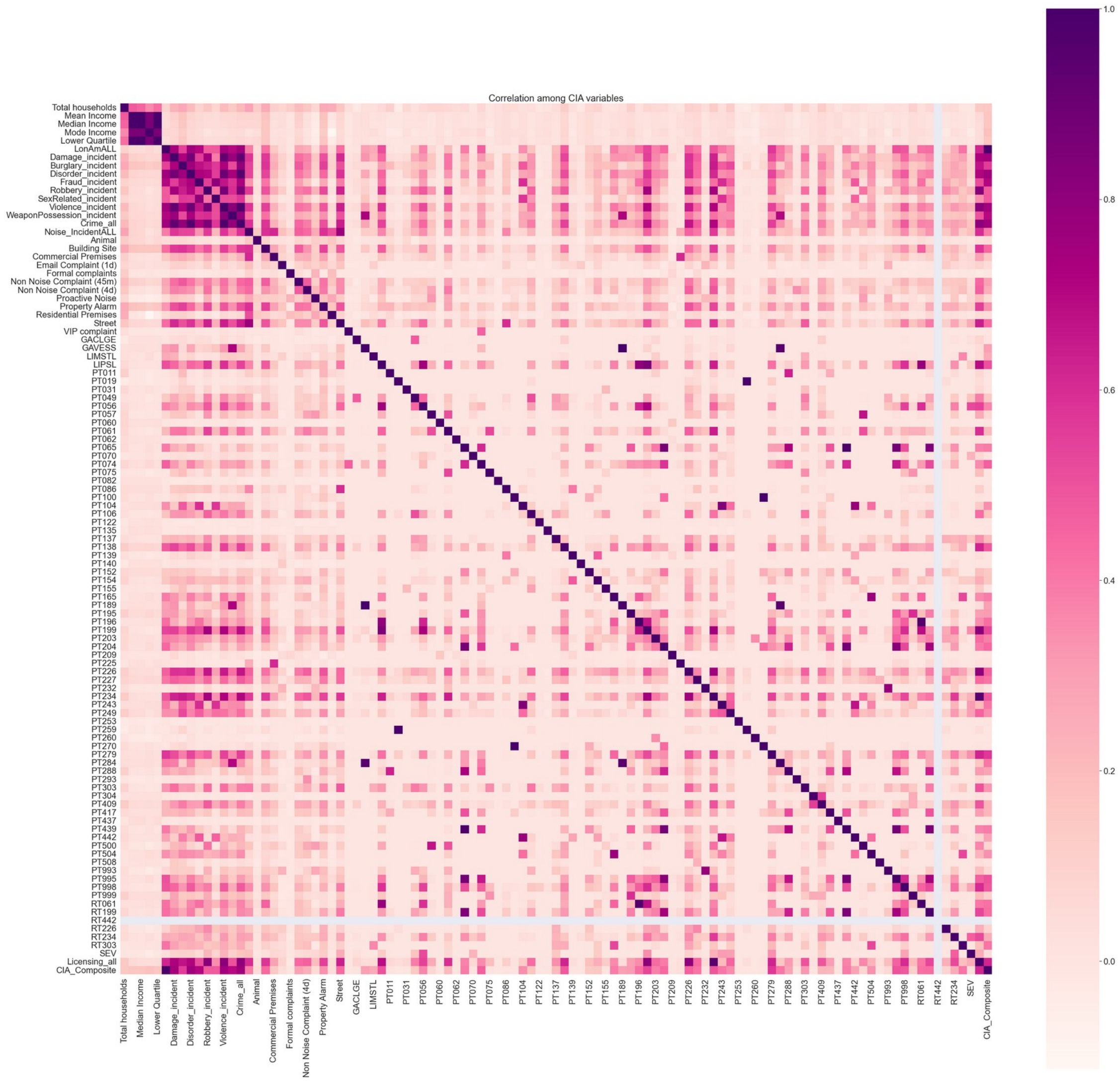
{"version_major":2,"version_minor":0,"model_id":"438b21e5e8104e16a7f1b
c6abc7db807"}

```python
CIA_corr = CIA.corr()
```

```python
plt.figure(figsize=(50,50))
sns.set(font_scale=2)
sns.heatmap(CIA_corr, vmax=1, square=True, annot=False,
annot_kws={"size": 20}, cmap='RdPu')


plt.title('Correlation among CIA variables')
```

```
Text(0.5, 1.0, 'Correlation among CIA variables')
```



```
wkhtmltopdf notebook.html notebook.pdf

  File
"/var/folders/c3/qc9fdqbj0lvdcbr2l68p4t2h0000gn/T/ipykernel_1449/69173
0344.py", line 1
    wkhtmltopdf notebook.html notebook.pdf
                ^
SyntaxError: invalid syntax
```