**1. Project overview:**

﻿The following dataset contains statistics in arrests per 100,000 residents for assault and murder, in each of the 50 US states, in 1973. Also given is the percentage of the population living in urban areas.

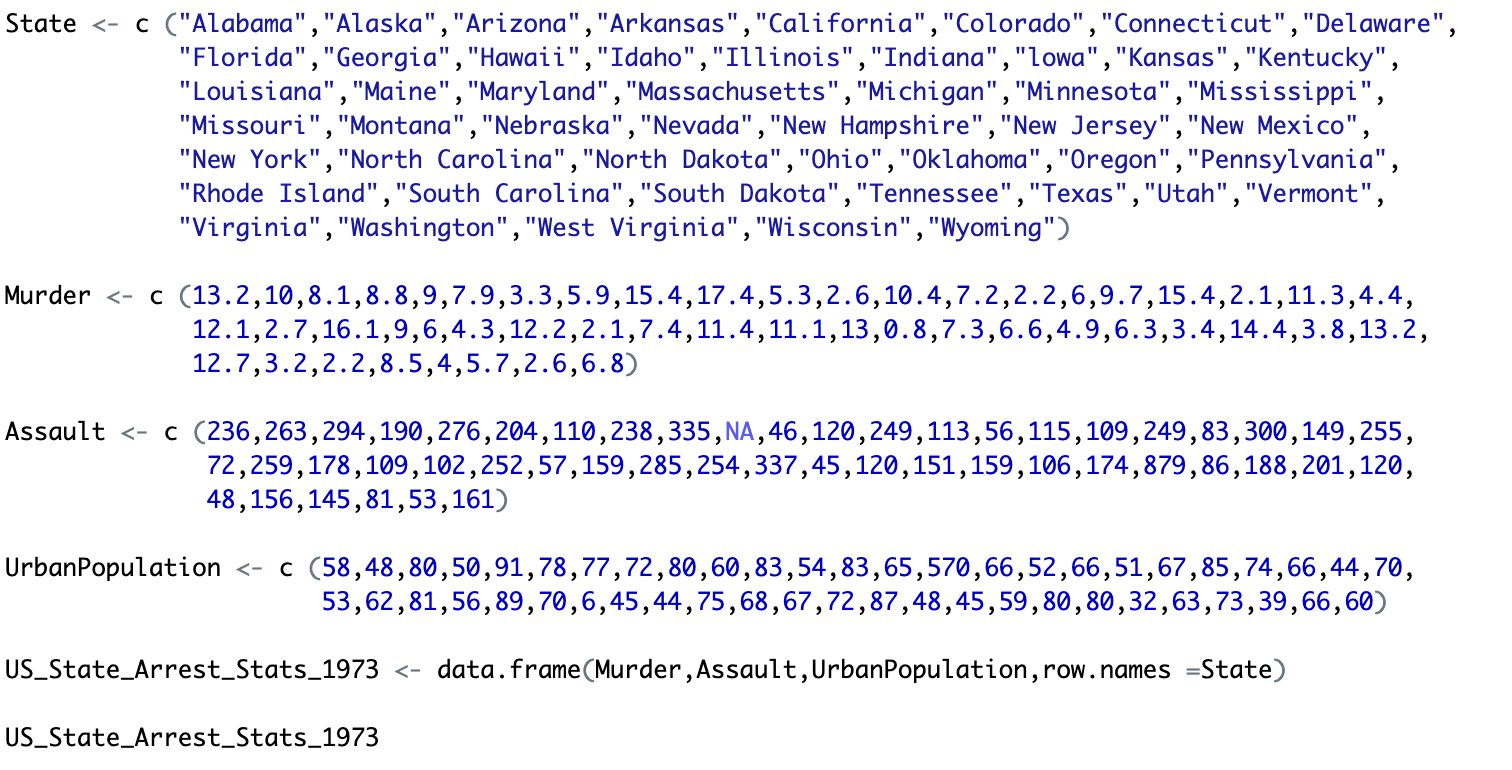
|  |  |  |  |
| --- | --- | --- | --- |
|  | Murder | Assault | Urban Population (%) |
| Alabama | 13.2 | 236 | 58 |
| Alaska | 10 | 263 | 48 |
| Arizona | 8.1 | 294 | 80 |
| Arkansas | 8.8 | 190 | 50 |
| California | 9 | 276 | 91 |
| Colorado | 7.9 | 204 | 78 |
| Connecticut | 3.3 | 110 | 77 |
| Delaware | 5.9 | 238 | 72 |
| Florida | 15.4 | 335 | 80 |
| Georgia | 17.4 |  | 60 |
| Hawaii | 5.3 | 46 | 83 |
| Idaho | 2.6 | 120 | 54 |
| Illinois | 10.4 | 249 | 83 |
| Indiana | 7.2 | 113 | 65 |
| lowa | 2.2 | 56 | 570 |
| Kansas | 6 | 115 | 66 |
| Kentucky | 9.7 | 109 | 52 |
| Louisiana | 15.4 | 249 | 66 |
| Maine | 2.1 | 83 | 51 |
| Maryland | 11.3 | 300 | 67 |
| Massachusetts | 4.4 | 149 | 85 |
| Michigan | 12.1 | 255 | 74 |
| Minnesota | 2.7 | 72 | 66 |
| Mississippi | 16.1 | 259 | 44 |
| Missouri | 9 | 178 | 70 |
| Montana | 6 | 109 | 53 |
| Nebraska | 4.3 | 102 | 62 |
| Nevada | 12.2 | 252 | 81 |
| New Hampshire | 2.1 | 57 | 56 |
| New Jersey | 7.4 | 159 | 89 |
| New Mexico | 11.4 | 285 | 70 |
| New York | 11.1 | 254 | 6 |
| North Carolina | 13 | 337 | 45 |
| North Dakota | 0.8 | 45 | 44 |
| Ohio | 7.3 | 120 | 75 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Murder | Assault | Urban Population (%) |
| Oklahoma | 6.6 | 151 | 68 |
| Oregon | 4.9 | 159 | 67 |
| Pennsylvania | 6.3 | 106 | 72 |
| Rhode Island | 3.4 | 174 | 87 |
| South Carolina | 14.4 | 879 | 48 |
| South Dakota | 3.8 | 86 | 45 |
| Tennessee | 13.2 | 188 | 59 |
| Texas | 12.7 | 201 | 80 |
| Utah | 3.2 | 120 | 80 |
| Vermont | 2.2 | 48 | 32 |
| Virginia | 8.5 | 156 | 63 |
| Washington | 4 | 145 | 73 |
| West Virginia | 5.7 | 81 | 39 |
| Wisconsin | 2.6 | 53 | 66 |
| Wyoming | 6.8 | 161 | 60 |

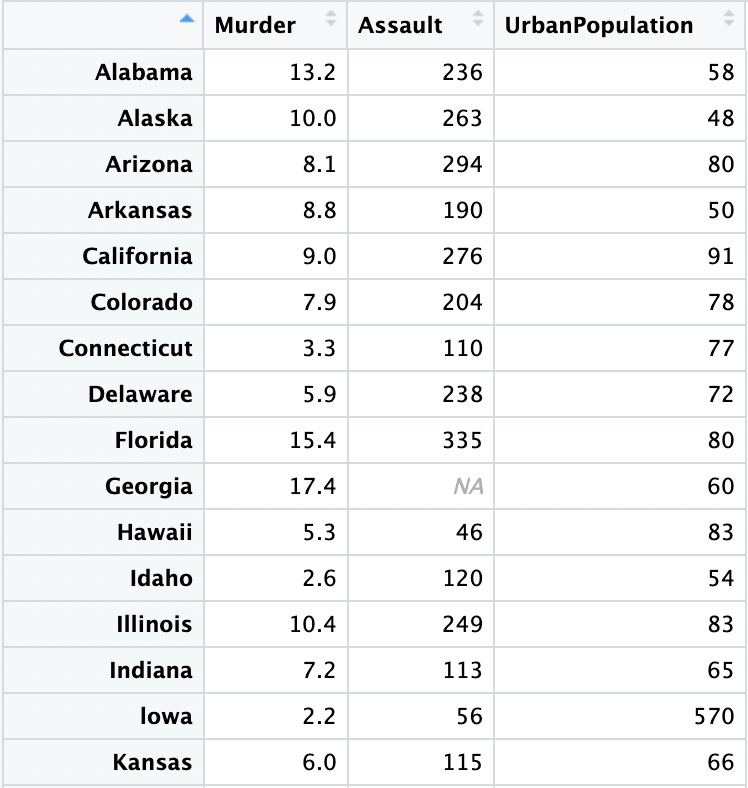
**2. Project Solution Design:**

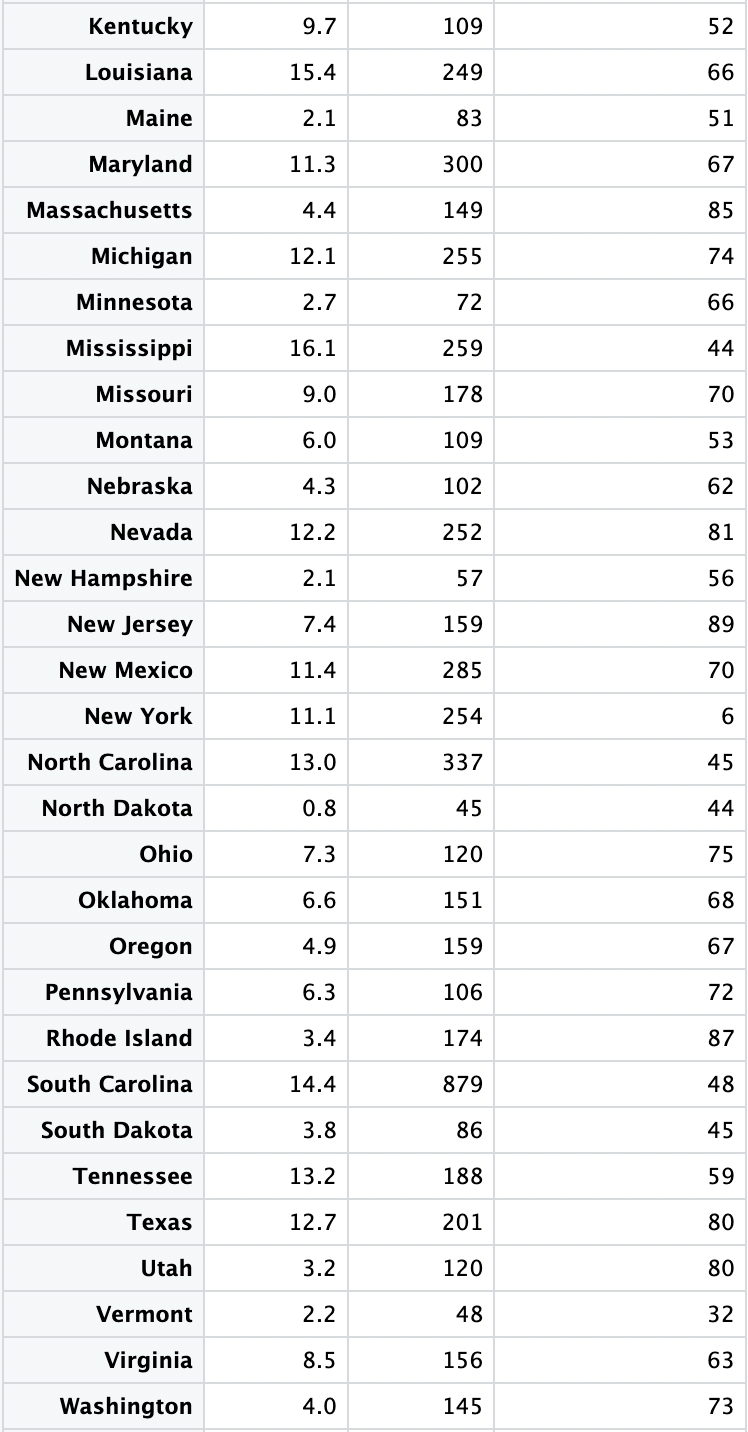
The objective of this project is to preprocess the dataset in order to prepare it for data analysis. Initially, a data frame was created by combining four vectors, namely "State", "Murder", "Assault", and "UrbanPopulation". The first step of data preprocessing involved replacing the missing data with the mean value of the respective column. Further, to smoothen the noisy data, the outliers were replaced with more appropriate values. In the data reduction phase, double or floating-point numbers were converted to integers. Subsequently, in the first step of data discretization, a categorical column was created based on the "UrbanPopulation" variable. In the second step, an ordered factor variable named "OrderedFactorPopulation" was created based on the "PopulationLevel" variable. These steps have been followed to ensure the dataset is in an appropriate format for data analysis.

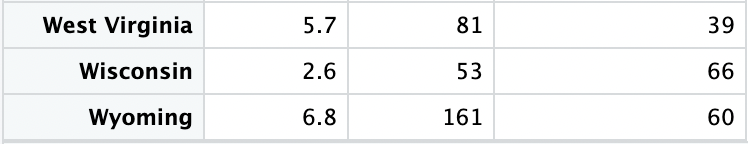
# **3. Data Frame:**

****

**Figure:** Code for Creating Data Frame







**Figure:** Initial Data Frame (US\_State\_Arrest\_Stats\_1973)

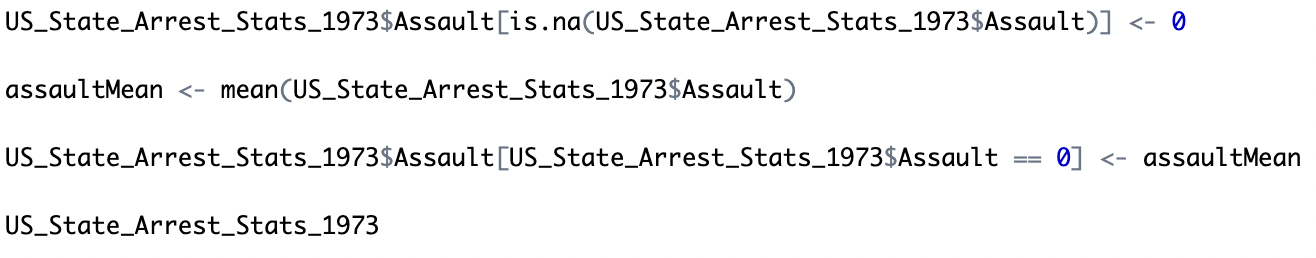
# **4. Data Preprocessing:**

## ﻿**4.1. Data cleaning:**

### **Handling Missing Data:**



We have a missing data at the “Assault” column of “Georgia” state. In order to replace the missing data, first it was replaced with 0 then we took the mean value of “Assault” column and replaced the value.



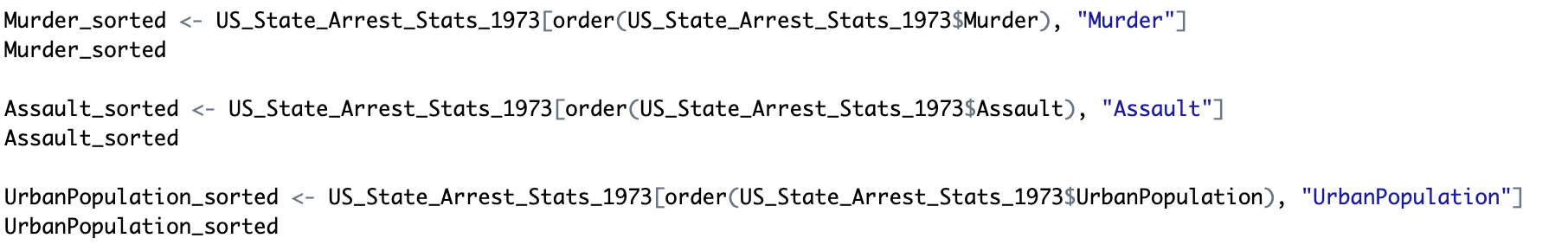
**Figure:** Code for Replacing the Missing Data



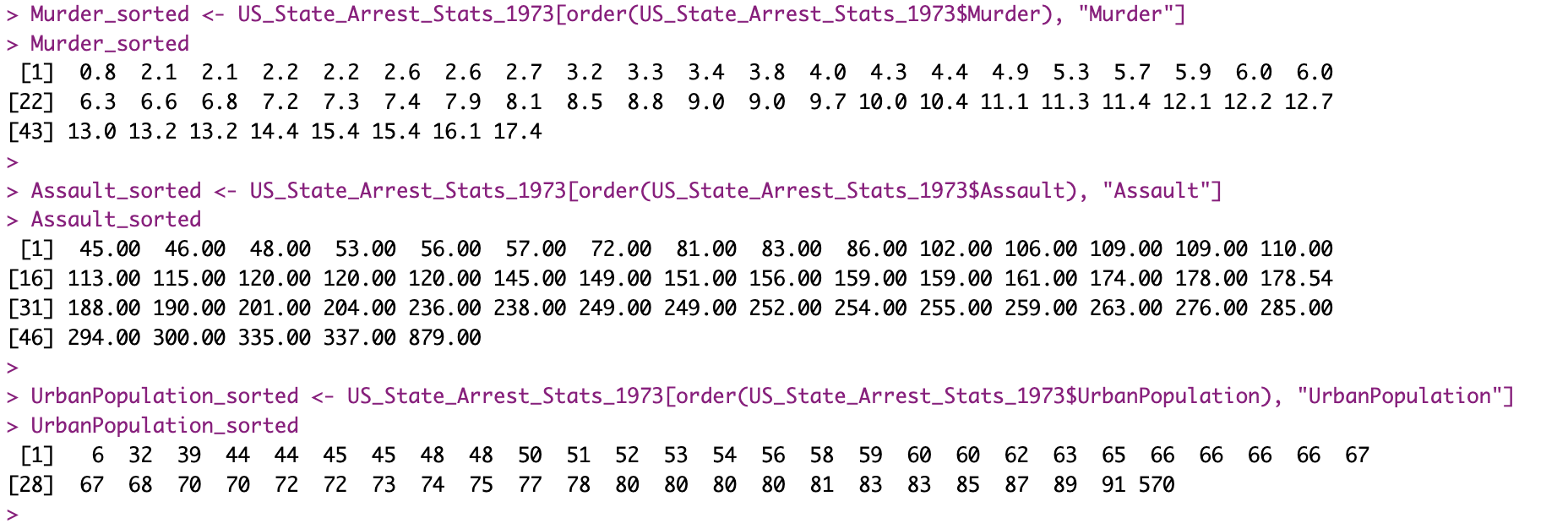
**Figure:** After Replacing the Missing Data

### **Smooth Noisy Data:**

In order to find the noisy data or the outlier we used order function to sort the data.



**Figure:** Code for Order Function to Find Outliers



**Figure:** Output of Order Function to Find Outliers

In the “Murder” column we don’t see any abnormal value which can be identified as outlier but in the “Assault” column we see value 879.00 which is pretty abnormal compared to other values of that column, same for the “Urban Population” column the value 6 is significantly low and 570 is significantly higher compared to other values.

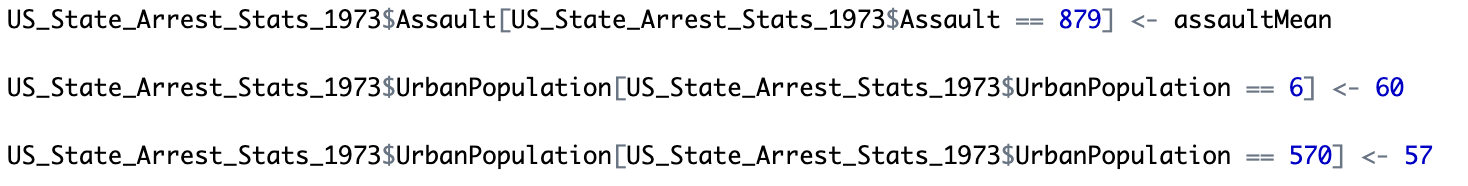






**Figure:** Sample with Outliers

To address the problem of outlier we have replace the value of “Assault” column 879.00 using the mean of the “Assault” column values. For the case of “UrbanPopulation” column the value 6 and 570 we assume there is some missing input for the value 6, so we replace 6 with 60 and for 570 we assume the 0 is extra since it cannot be more than 100 so we replace 570 with 57.



**Figure:** Code for Replacing Outliers







**Figure:** After Replacing Outliers

### **Data Wrangling or Munging:**

Fortunately for this dataset no wrangling or munging is needed since all data are in same pattern.

## **4.2. Data Integration:**

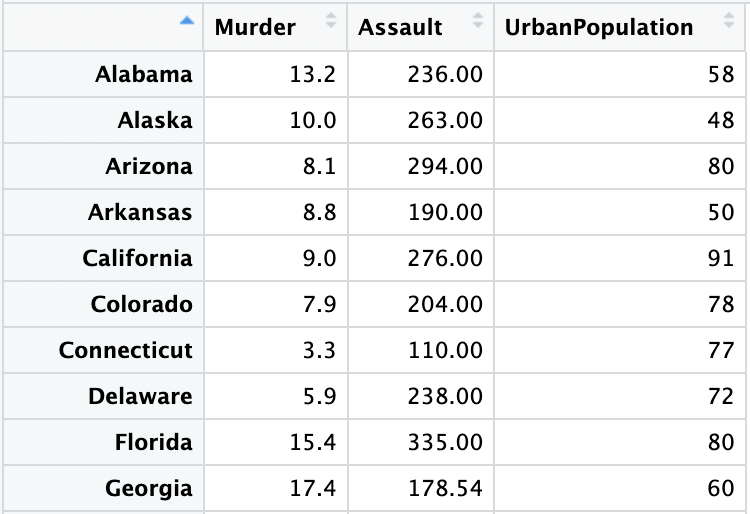
For this project no data integration is needed as we collected the data from a single source.

## **4.3. Data Transformation:**

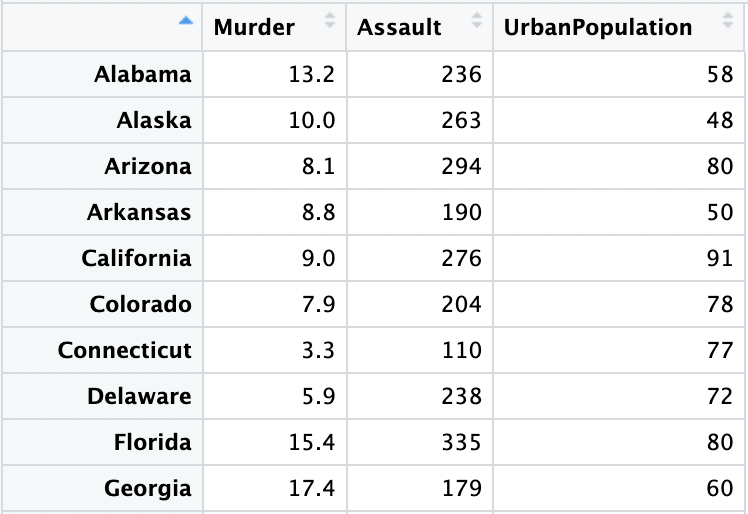
The steps of data transformation are already performed during smoothing noisy data. So, transformation is not required anymore.

## **4.4. Data Reduction:**

The values of “Assault” column was integer value before replacing the missing value with the mean value but now they are float or decimal so we can reduce the size by ceiling those value with no value after the decimal point.

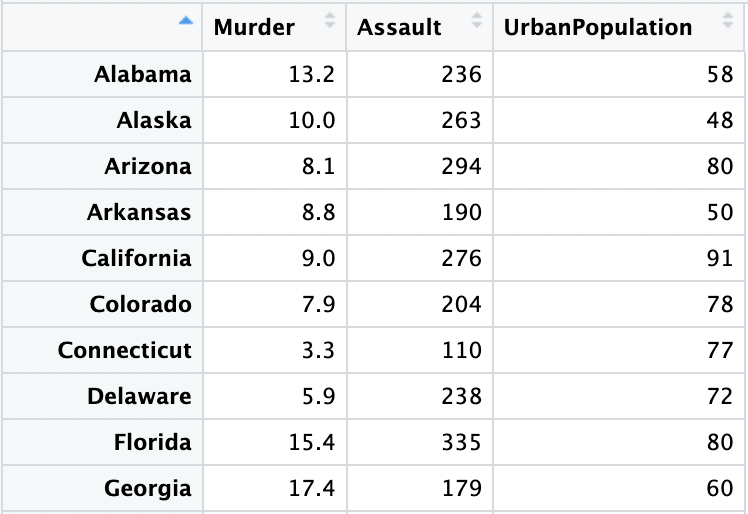


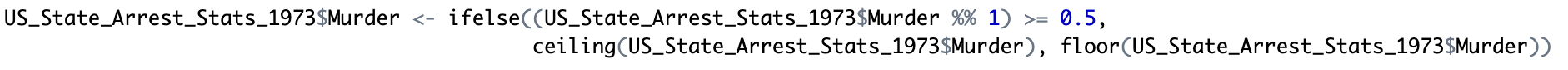


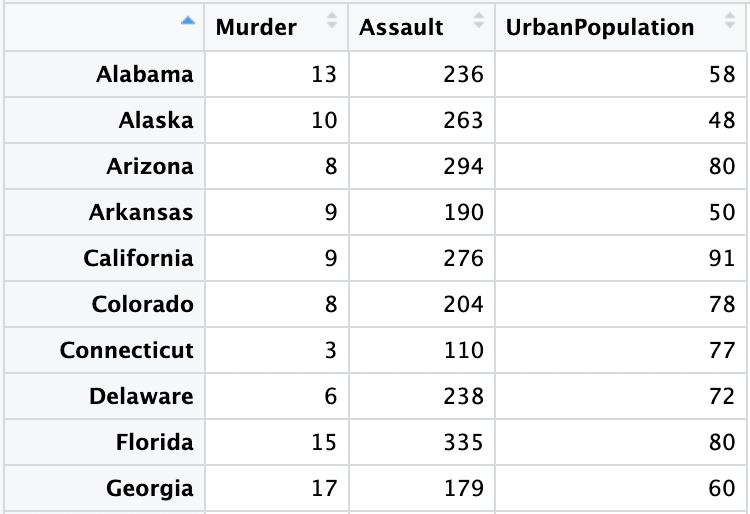


**Figure:** Before and After the Data Reduction of Assault Column Using Ceiling Function

Similarly, for the “Murder” column we used ceiling function if the value is greater or equal to 5 after the decimal point and used floor function if the value is less than 5 after the decimal point.



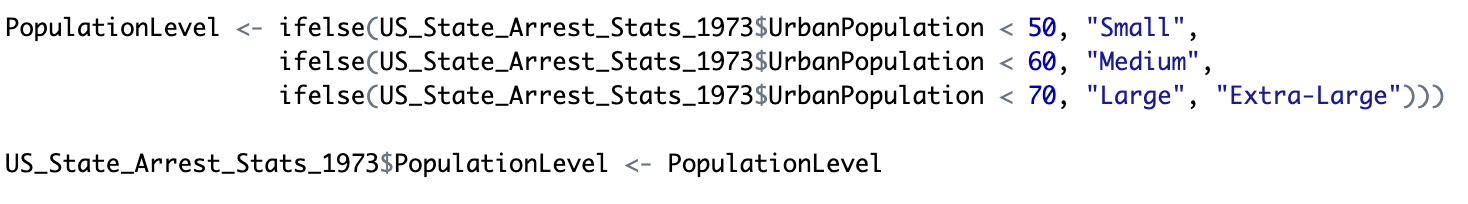


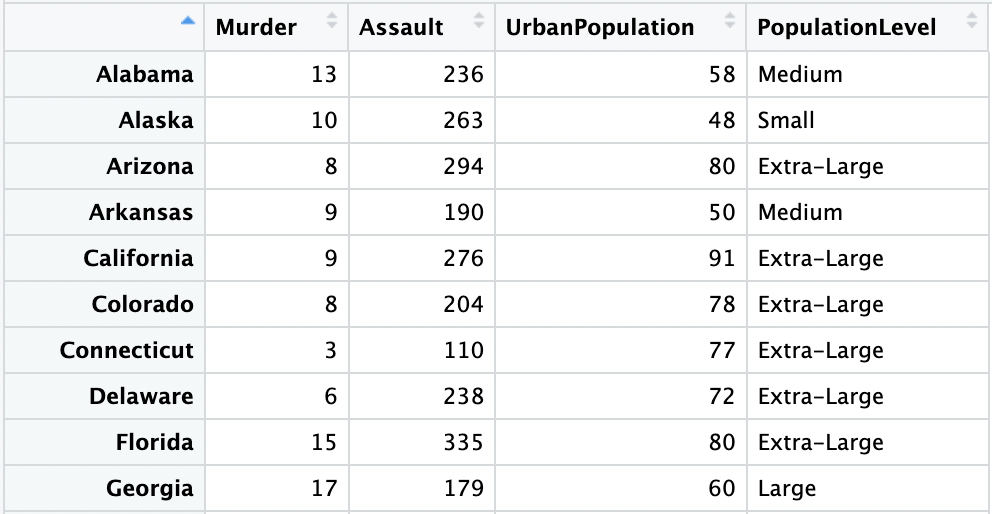


**Figure:** Before and After the Data Reduction of Murder Column Using Ceiling and Floor Function

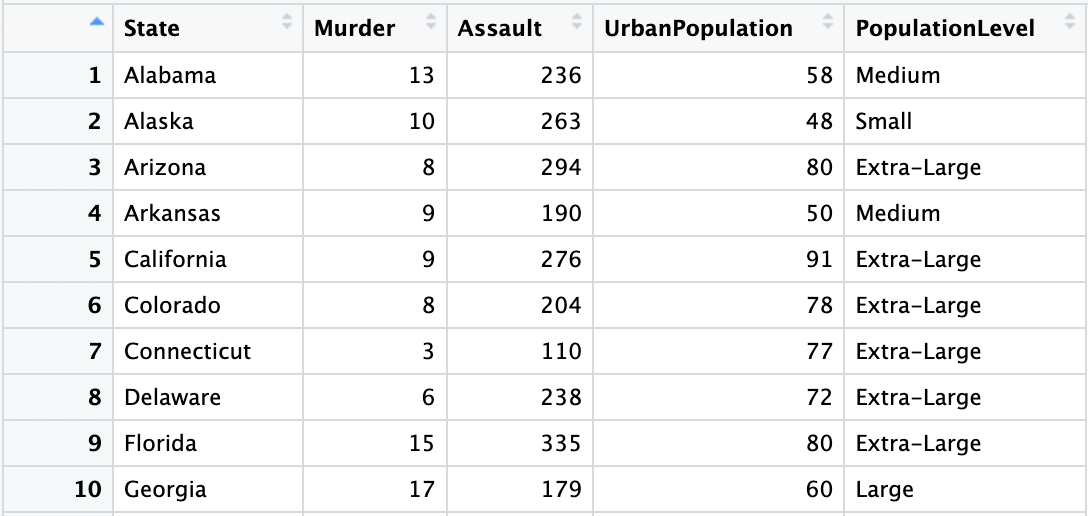
## **4.5. Data Discretization:**

In this part of the project, we added two new columns. We created the first column “PopulationLevel” with the help of “UrbanPopulation”. ﻿Convert the urban population percentage into level, small (<50%), medium (>= 50% to <60%), large (>= 60 to <70%), and extra-large (70% and above) to create the column with categorical data.

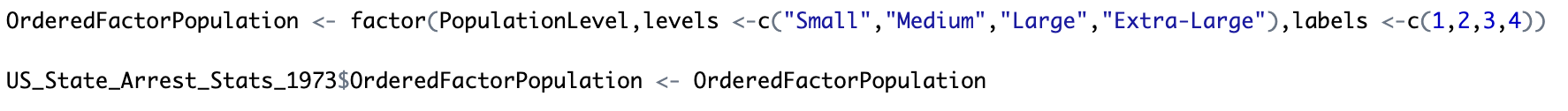


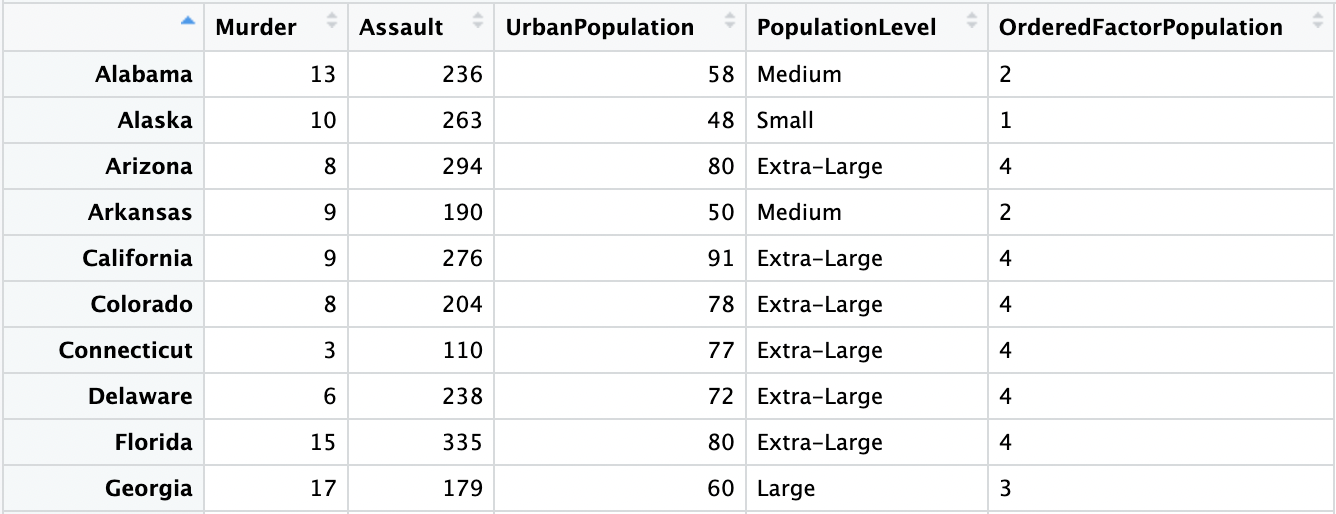


**Figure:** First 10 Rows After Adding PopulationLevel Column



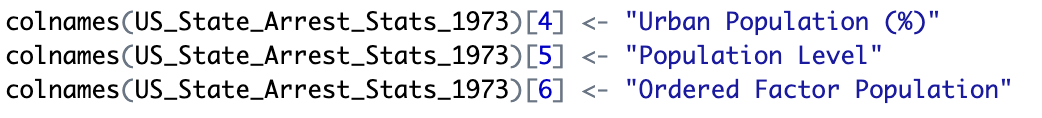
﻿For the second column. Since, the data of the column “PopulationLevel” are categorical, we create another column named “OrderedFactorPopulation” based on “PopulationLevel” which is an ordered factor variable. Where, Small = 1, Medium = 2, Large = 3, and Extra-large = 4.



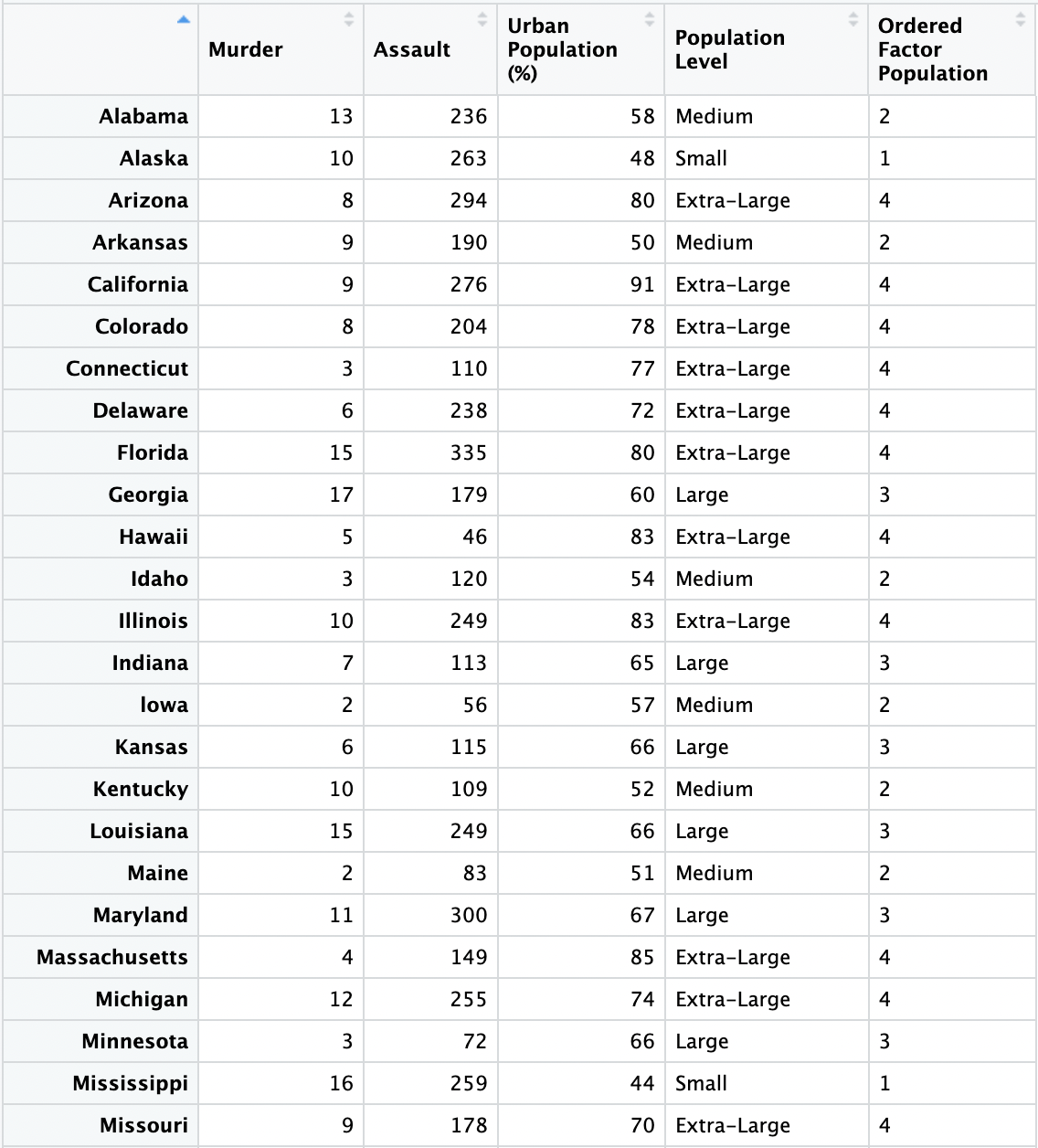


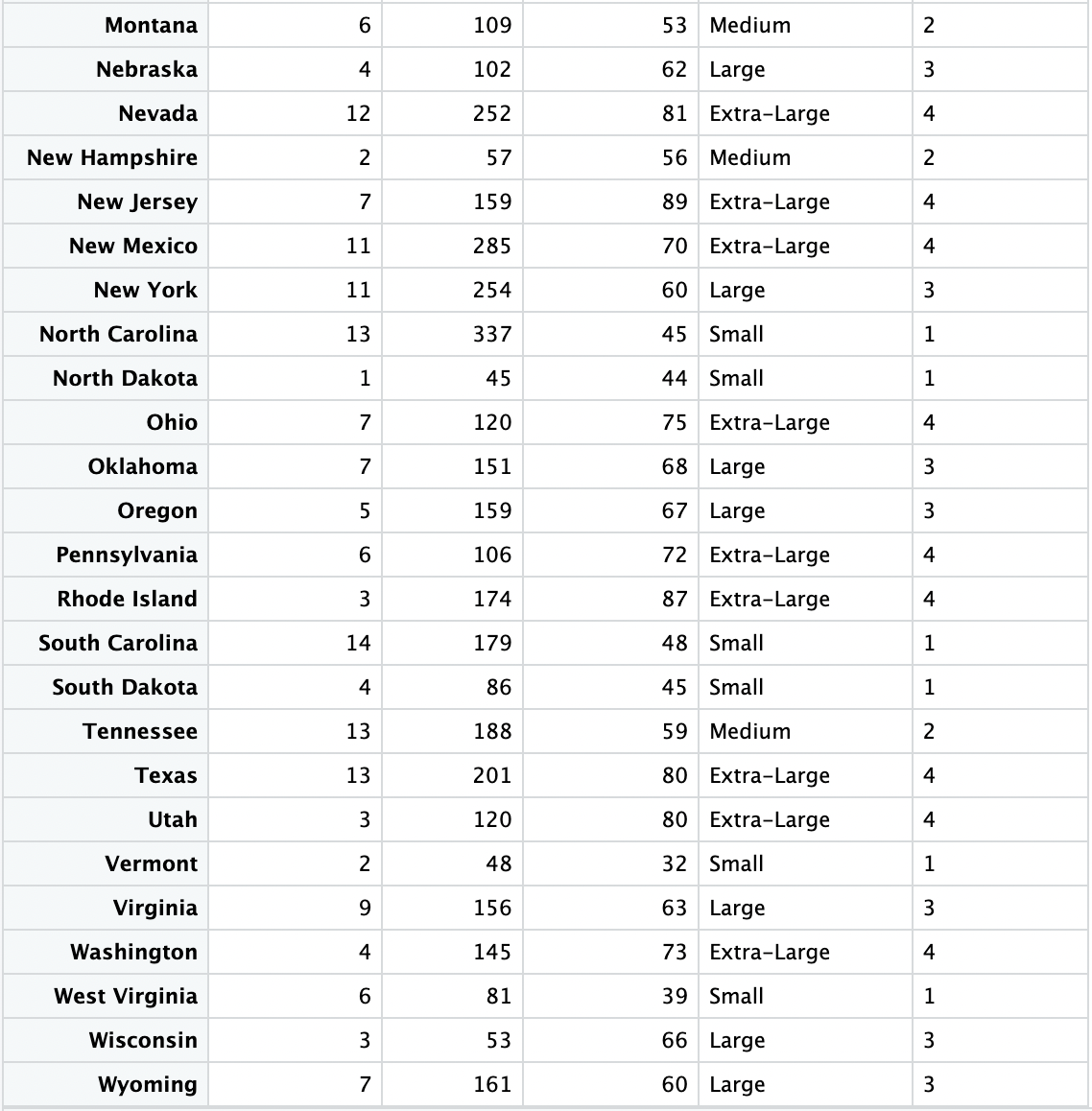
**Figure:** First 10 Rows After Adding OrderedFactorPopulation Column

We change some of the column’s name to match the given name in the project with colnames function.



# **The Cleaned Dataset:**





**Figure:** FinalDataset After Preprocessing