**Project overview:**

Data pre-processing techniques are used when the data is inconsistent, which indicates that the data is not recorded in accordance with the restrictions on the column, noisy, which may contain a variety of mistakes or outliers, and incomplete, which indicates that some attribute value is missing. In our given dataset initially, I have seen some missing value. After handle missing value it may occurs some noise like format problem. Then I have fixed format with using my preferable approach. After changing format, I have added a new column using another column value what had been given in our question condition. After adding column then I tried to handle categorical value to numerical value what is our discretization part. And finally, I tried to compare 3 columns value using normalization technique.

**Project solution design:**

Import Data Set

Data Cleaning (Missing Data replace)

Data Integrating

Data Reduction

Data transformation (Normalization)

Data Discretization

Fig: Block Diagram Of the project Solution

**Data pre-processing:**

**1. Import dataset**: Before we start preprocessing our data, first we need to load or import our data. We can import our data using > dataset<- read.csv( “DataSetR.csv”) syntax. After import we can see using Just Command >dataset

Code: > dataset<- read.csv( “DataSetR.csv”)

>dataset

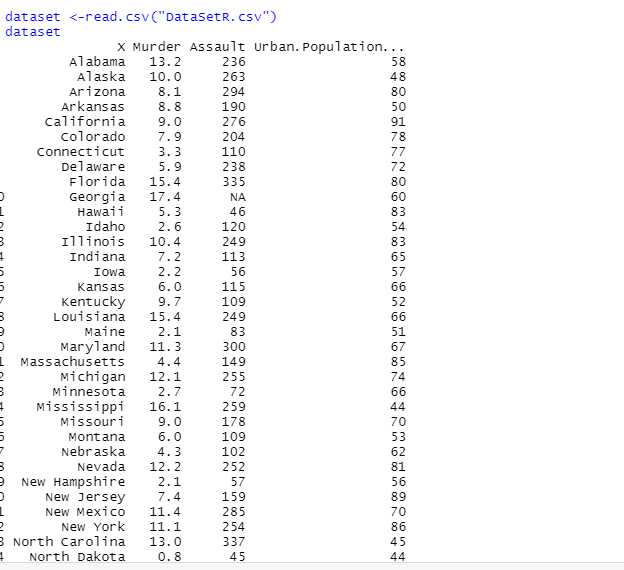


Fig: After import .csv File

**2.Data Cleaning:**

2.1 **Handling the missing data:** The Assault column value in the dataset has missing data. This issue must be resolved before to incorporating a data set into a model; otherwise, it will seriously impact that model. So we should handle this data set. we can handle 2 ways either Replace the data or Discard. As it has only 50 data so we couldn’t remove any data. so we can replace using average function as a numerical data.

Code:

>dataset$Assault = ifelse(is.na(dataset$Assault), mean(dataset$Assault, na.rm = TRUE), dataset$Assault)

>dataset

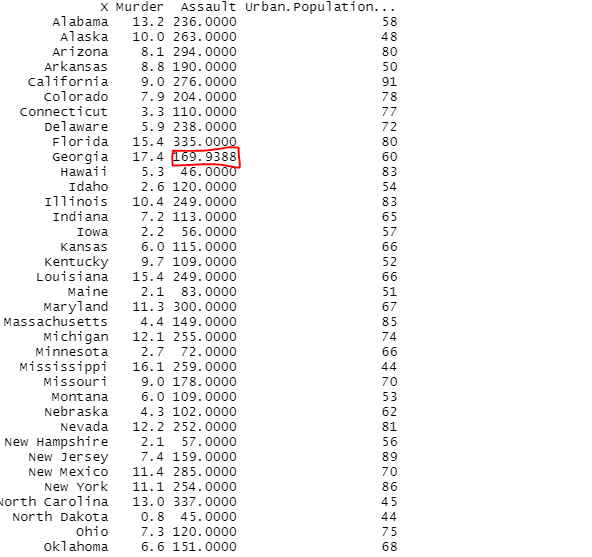


Fig: After replace missing value

3. **Data Integration:** Data integration is a process where we need to integrate new data from different source or table Let's now imagine that we must add a new column of data to our data table based on estimates of the urban population. Here is my code to add new column name “type”

Code:

> dataset$type <- rep(NA, nrow(dataset))

> dataset[dataset$Urban.Population...>0 & dataset$Urban.Population...< 50, ][, "type"] <- "small"

> dataset[dataset$Urban.Population...>=50 & dataset$Urban.Population...<60, ][, "type"] <- "medium"

> dataset[dataset$Urban.Population...>=60 & dataset$Urban.Population...<70, ][, "type"] <- "large"

> dataset[dataset$Urban.Population... >=70, ][, "type"] <- "extra large"

> dataset

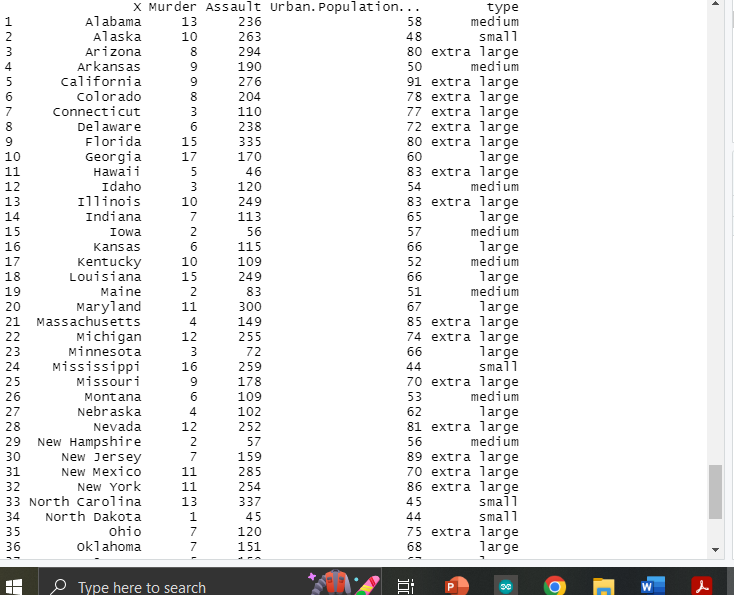


Fig: After integrate a new column

**4. Data transformation:**

As has already been known, the data transformation process includes one or more of the following steps: normalization, summarization, noise removal, smoothing, and summarizing of the data. for our data set I used normalization.

4.1 **Normalization:**

The statistical distribution of the data is positively impacted by normalization procedures since they allow us to minimize the magnitude of the variables.in this data set I have normalized column between 2 to 4.

Code:

> min\_max\_norm <- function(x) { (x - min(x)) / (max(x) - min(x)) }

> dataset <- as.data.frame(lapply(dataset[2:4], min\_max\_norm))

> dataset

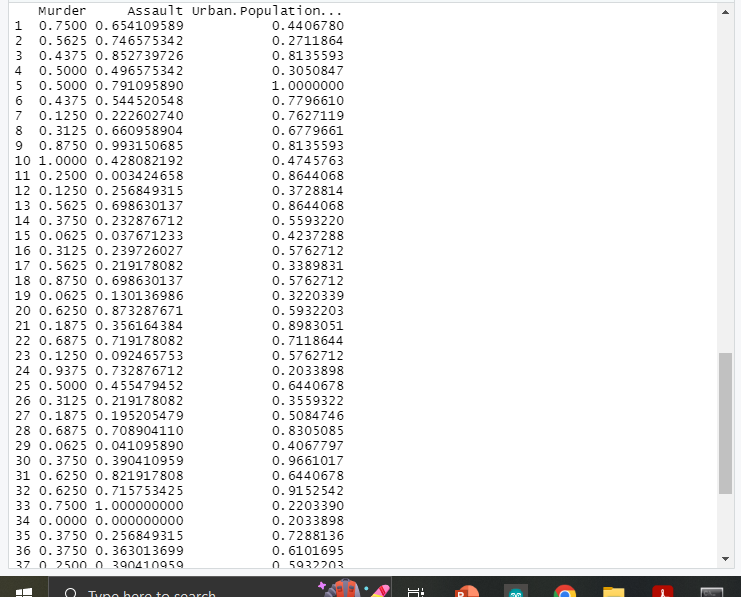


Fig: After Normalized Data table

**5.Data Reduction:**

Given the enormous number of rows in this dataset, it could be more logical to round the murder and assault rates per capita to the nearest zero decimal places, depending on the processing and storage resources we have available.

Code: dataset$ Assault = as. numeric(format(round(dataset$ Assault ,0)

dataset$Murder = as. numeric(format(round(dataset$ Murder ,0)

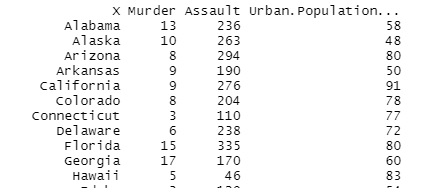


Figure: After Reducing value size

**6.** **Data discretization:**

Data discretization is a one kind of reducing process.in can be categorical to numerical. In our dataset ‘type’ column had four categorical value. which I have replace 4 numerical value:

“small” replace by 0. “medium” replace by 1. “large” replaced by 2 “extra large” replaced by 3

Code:

> dataset$type = factor(dataset$type,levels = c('small', 'medium','large','extra large'), labels = c(0,1,2,3))

> dataset

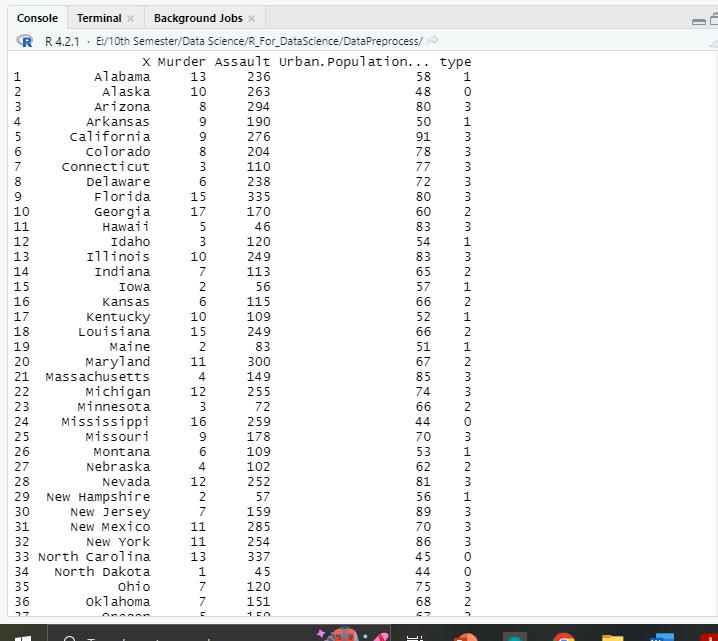


Fig: After discretization

**Discussion and Conclusion:** This is the end of our data preprocessing for our given dataset. I hope now it’s clean and well prepared to give any models that can predict future outcomes effectively