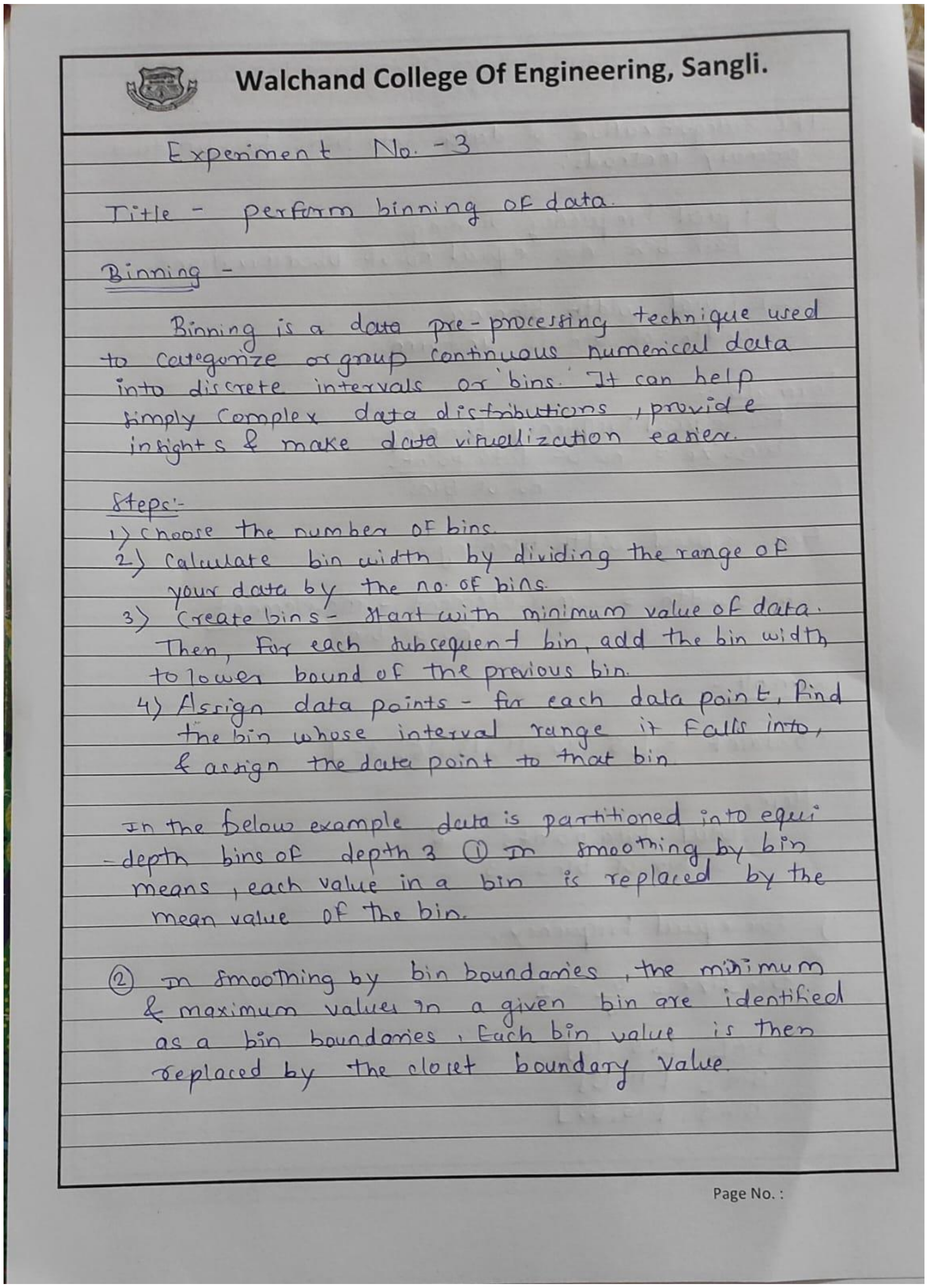
Experiment no. 3

Name: Sonali Dattatray Kaingade

PRN: 21620002

Title: To perform Binning of data



Walchand College Of Engineering, Sangli.

Experiment No. -3

Title - perform binning of data.

Binning -

Binning is a data pre-processing technique used to categorize or group continuous numerical data into discrete intervals or bins. It can help simply complex data distributions, provide insights & make data visualization easier.

Steps:-

1) Choose the number of bins.
2) Calculate bin width by dividing the range of your data by the no. of bins.
3) Create bins - start with minimum value of data. Then, For each subsequent bin, add the bin width to lower bound of the previous bin.
4) Assign data points - for each data point, find the bin whose interval range it falls into, & assign the data point to that bin.

In the below example data is partitioned into equi-depth bins of depth 3 ① In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.

② In smoothing by bin boundaries, the minimum & maximum values in a given bin are identified as a bin boundaries, Each bin value is then replaced by the closet boundary value.

Page No. :

The Categorization of data into bins follows two primary methods:

1) Equal Frequency binning
Each bin has equal no. of bberservations.

2) Equal width binning
width of bin is uniform.

Formula -

For equal Frequency binning,

$$\text{bin\_size} = \frac{\text{no. of data points}}{\text{no. of bins.}}$$

For equal width binning.

$$\text{bin\_width} = \frac{\text{max\_element} - \text{min\_element}}{\text{no. of bins}}$$

Example -

Dataset - [5, 10, 11, 13, 15, 35]

Here,
total data points = 6
No. of bins = 3

1) For equal Frequency,

$$\text{bin\_size} = \frac{6}{3} = 2$$

∴ Bin1 [5, 10]
Bin2 [11, 13]
Bin3 [15, 35]

2) For equal width binning,

$$\text{bin width} = \frac{35-5}{3} = \frac{30}{3} = 10$$

$\therefore$ Bin 1 = [5, 10, 11, 13]
Bin 2 = [15]
Bin 3 = [35]

Conclusion:
Binning is useful technique for transforming continuous data into discrete categories, making it easier to analyze & visualize.

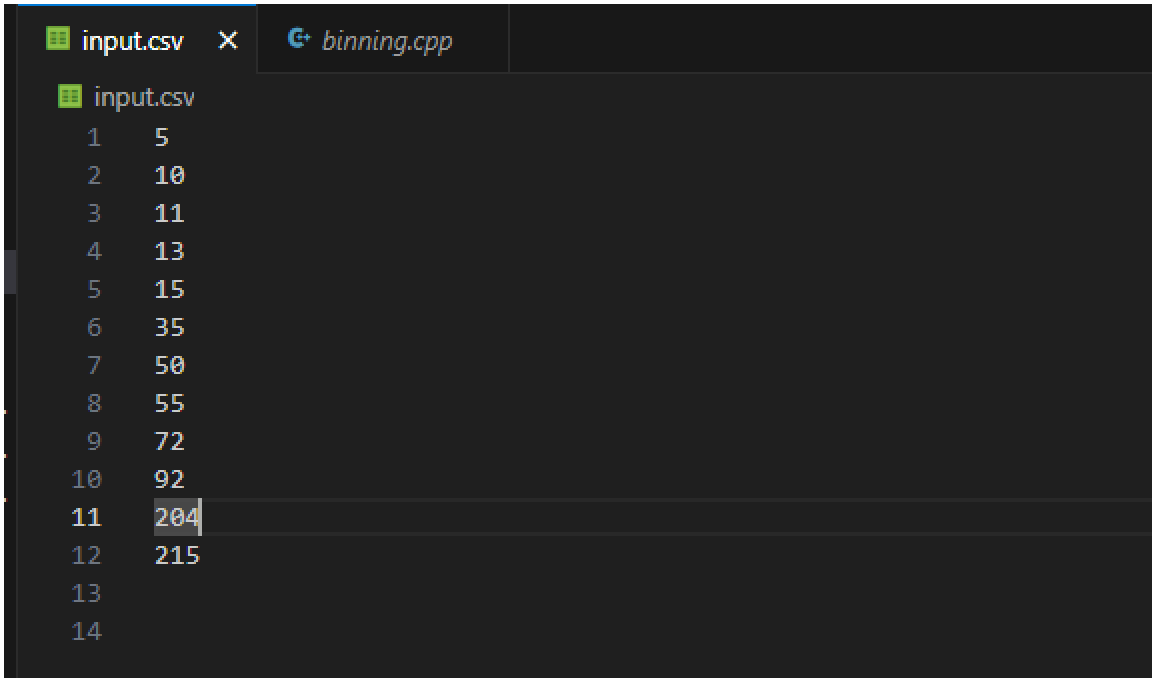
Code:

```
#include <iostream>
#include <fstream>
#include <vector>
#include<algorithm>
#include <climits>
#include<cmath>
using namespace std;
//equal frequency
vector<vector<int>> equi_frequency(vector<int> data,double m)
{

    double n=data.size();
    double ele=ceil(n/m);

    vector<vector<int>> totalbins;
    for(int i=0;i<m;i++)
    {
        vector<int> bin;
        for(int j=i*ele;j<(i+1)*ele;j++)
        {
            if(j>=n)
            {
                break;
            }
            bin.push_back(data[j]);
        }
        totalbins.push_back(bin);
    }
    return totalbins;
}

//equal width
vector<vector<int>>equi_width(vector<int> data,int m)
{
    int n=data.size();

    int min_ele=INT_MAX;
    int max_ele=INT_MIN;

    for(int i=0;i<data.size();i++)
    {
        min_ele= min(min_ele,data[i]);
        max_ele= max(max_ele,data[i]);
    }
    int w = (max_ele-min_ele)/m;
    int min1=min_ele;

    vector<int> arr;
    for(int i=0;i<m+1;i++)
    {
        arr.push_back(min1+w*i);
    }

    vector<vector<int>> arri;

    for(int i=0;i<m;i++)
    {
        vector<int> temp;
        for(int k:data)
        {
            if(k>=arr[i] && k<=arr[i+1])
            {
                temp.push_back(k);
            }
        }
        arri.push_back(temp);
    }
return arri;
}

// Write binning outputs to CSV
void writeCSV(string filename, vector<vector<int>> bins)
{
  ofstream outputFile(filename);
  for (int i = 0; i < bins.size(); i++)
  {
    outputFile << "Bin " << i + 1 << ":"<<" ";
    for (int num : bins[i])
    {
      outputFile << num << ",";
    }
    outputFile << "\n";
  }
  outputFile.close();
}
```

```cpp
int main()
{

    ifstream inputf("input.csv");


    vector<int> data;
    int val;
    while(inputf>>val)
    {
        data.push_back(val);
    }
    sort(data.begin(),data.end());
     int method,m;
  cout << "Choose binning method: " << endl;
  cout << "1. Equal Frequency Binning" << endl;
  cout << "2. Equal Width Binning" << endl;
  cout << "\nEnter method number: ";
  cin >> method;
  cout << "\nEnter number of bins: ";
  cin >> m;

  if (method == 1)
  {
    vector<vector<int>> freqBins = equi_frequency(data, m);
    writeCSV("output_equi_frequency.csv", freqBins);
  }
  else if (method == 2)
  {
    vector<vector<int>> widthBins = equi_width(data, m);
    writeCSV("output_equi_width.csv", widthBins);
  }
  else
  {
    cout << "Invalid method choice." << endl;
  }


    return 0;

}
```

Output:

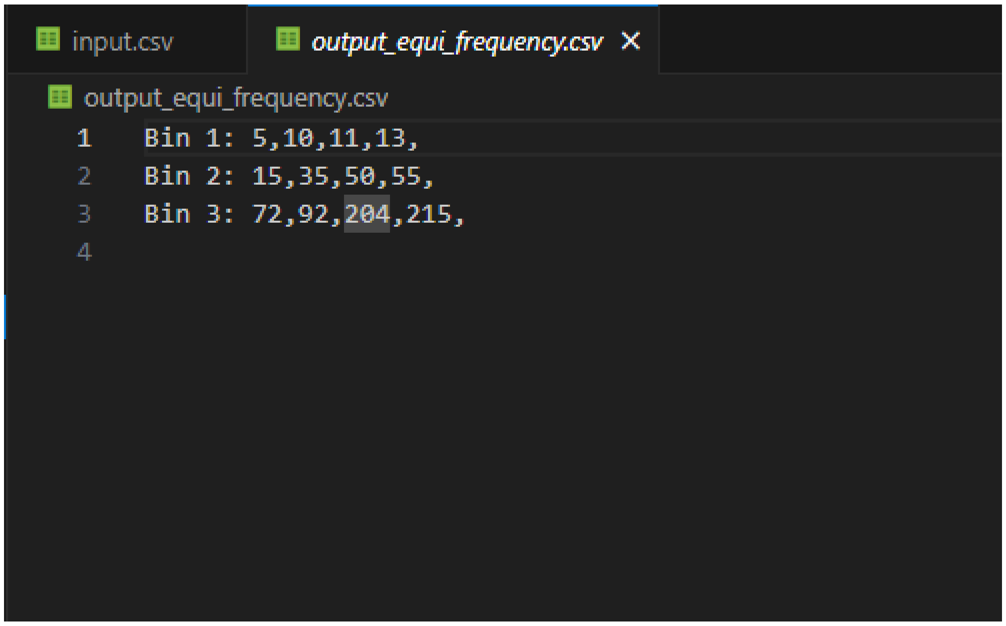
Input.csv



```
5
10
11
13
15
35
50
55
72
92
204
215
```

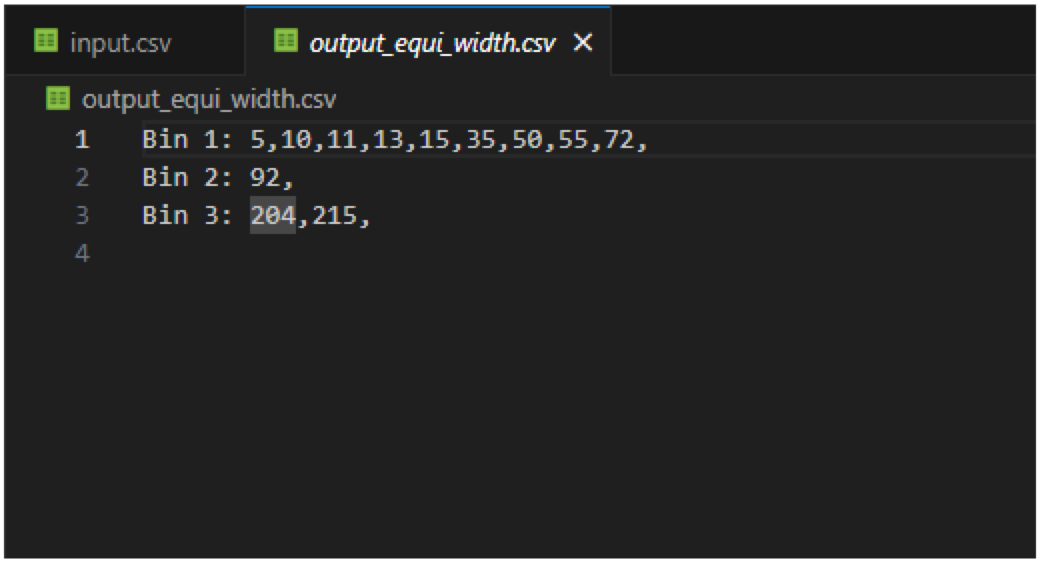
Output.csv

1.Equal frequency



```
Bin 1: 5,10,11,13,
Bin 2: 15,35,50,55,
Bin 3: 72,92,204,215,
```

2. Equal width



```
Bin 1: 5,10,11,13,15,35,50,55,72,
Bin 2: 92,
Bin 3: 204,215,
```