Introduction to Machine Learning Assignment 1

Name: Vivekanand Reddy Malipatel
CWID: A20524971

Question 1 Answers:

a. (The Python code for this is in the file: 1_a.py)

Output Screenshots :

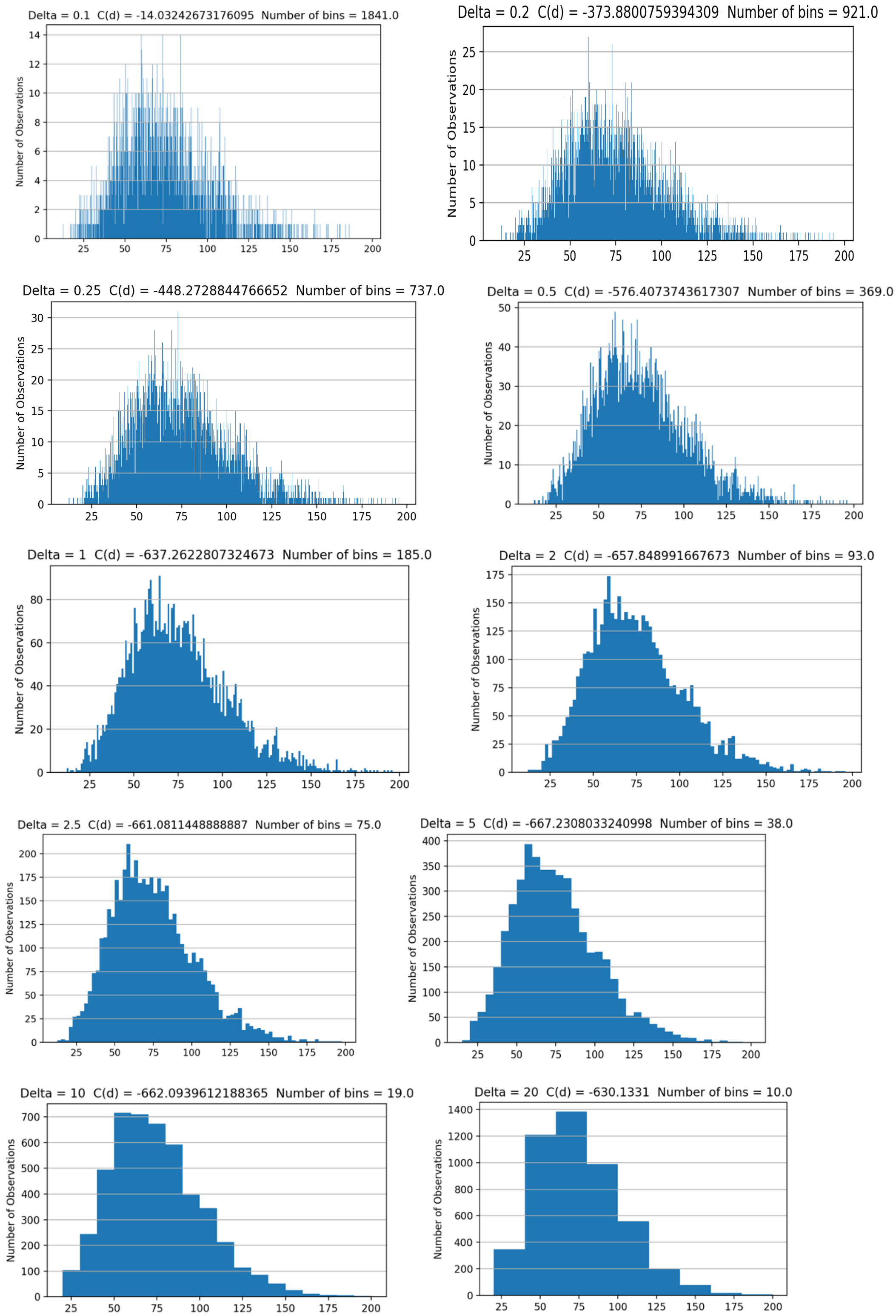
```
Count:  4804
Mean:  75.0568006
Standard Deviation:  27.4424988
Minimum:  11.55
25th Percentile:  55.46
Median:  71.825
75th Percentile:  91.1725
Maximum:  195.6
```

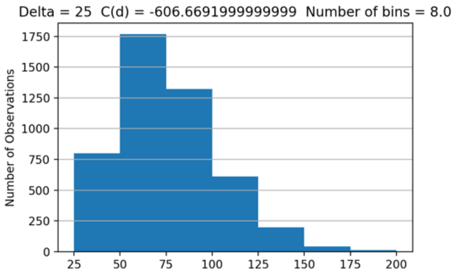
Answer:

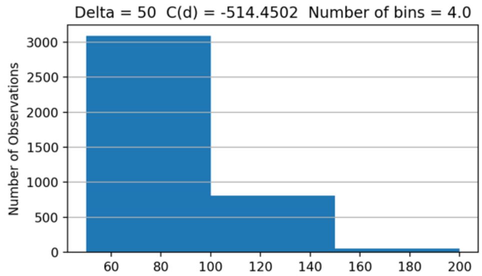
Count: 4804
Mean: 75.0568006
Standard Deviation: 27.4424988
Minimum: 11.55
25th Percentile: 55.46
Median: 71.825
75th Percentile: 91.1725
Maximum: 195.6

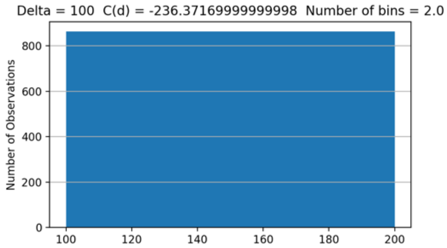
b. (The Python code for this is in the file: 1_b.py)

Output Histograms:


Delta = 0.1 C(d) = -14.03242673176095 Number of bins = 1841.0
Delta = 0.2 C(d) = -373.8800759394309 Number of bins = 921.0
Delta = 0.25 C(d) = -448.2728844766652 Number of bins = 737.0
Delta = 0.5 C(d) = -576.4073743617307 Number of bins = 369.0
Delta = 1 C(d) = -637.2622807324673 Number of bins = 185.0
Delta = 2 C(d) = -657.848991667673 Number of bins = 93.0
Delta = 2.5 C(d) = -661.0811448888887 Number of bins = 75.0
Delta = 5 C(d) = -667.2308033240998 Number of bins = 38.0
Delta = 10 C(d) = -662.0939612188365 Number of bins = 19.0
Delta = 20 C(d) = -630.1331 Number of bins = 10.0
Number of Observations







Answer :

if we want to choose a bin width from the given array of widths with the number of bins to be between 10 and 100 inclusively, the Shimazaki and Shinomoto (2007) method suggests to determine the width d that minimises C(d).

Hence, The Optimal Bin width would be **d=5**, with minimal **C(d) = -667.2308** (rounded value) which as **38 bins**.

Calculations for the recommended Bin Width, d=5:

Minimum of the given Data = 11.55
Maximum of the given Data = 195.6
Mean of the given data, $\bar{y}$ = 75.05680058284763

Specifying Bin Boundaries,
Rounding $\bar{y}$ with the integral multiple of the bin width d,

$$b_0 = 5 * round\ (75.05680058284763/5) = 75.0$$

Hence, boundary of central bin b_0 = 75.0

Number of bins on the left = round ((75.0-11.5) / 5) = 13.0
Number of bins on the right = round ((195.56 - 75.0) / 5) = 25.0

*left boundary of the first bin = 75.0-(13.0*5) = 10.0*
right boundary of the first bin = 10.0+5 = 15

Next step is to Add the Data range to the first bin.

Continuing the above three steps for the whole data by incrementing subsequent rightmost bin boundaries by the delta = 5.

The number of observations in subsequent 38 bins are,
[4,6,42,62,94,149,222,273,323,393,368,342,342,332,327,265,219,178,181,164,126,87,53,61,48,37,29,22,16,10,9,3,6,2,4,2,2,1]

Calculating the mean and variance of the number of observations,

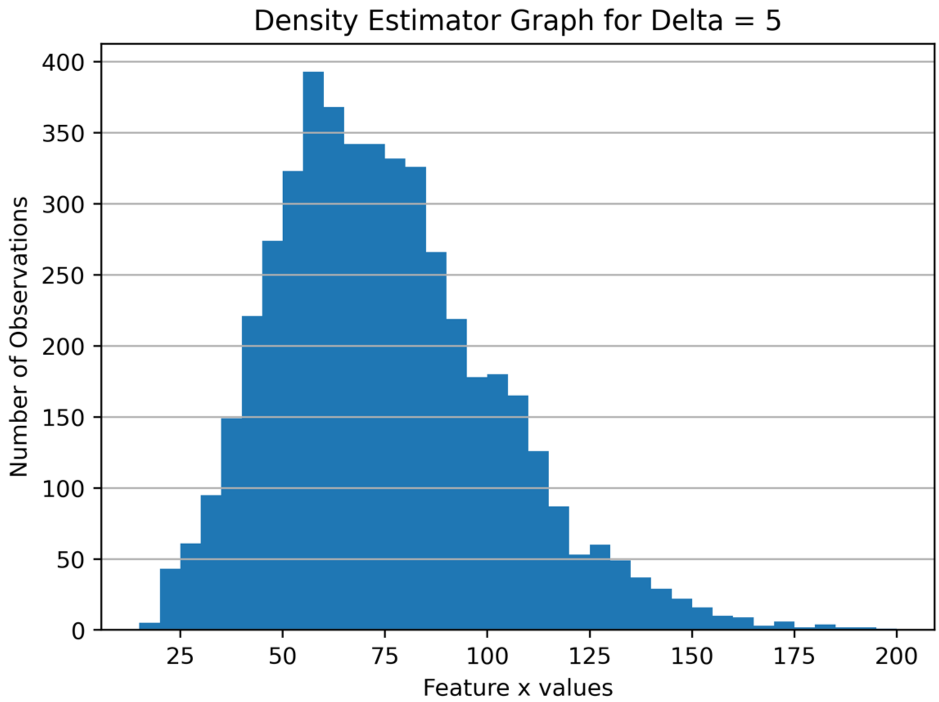
$$\bar{n} = mean(observations) = 126.42105263157895$$

$$v = mean((n_i - \bar{n})^2) = 16933.612188365652$$

Calculating *C(d) = (2*126.42105263157895-16933.612188365652) / 25*

Hence *C(d)* = **-667.2308033240998**

c. (The Python code for this is in the file: 1_c.py)



Question 2 Answers:

a. (The Python code for this is in the file: 2_a.py)

Output Screenshot :

```
Number of Observations : 5960

Frequency distributions of BAD :
0    4771
1    1189
Name: BAD, dtype: int64
Missing Values : 0

DEBTINC
Mean : 33.779915348721126
Standard Deviation : 8.60174618632853

LOAN
Mean : 18607.96979865772
Standard Deviation : 11207.480416694003

MORTDUE
Mean : 73760.817199559
Standard Deviation : 44457.60945841593

VALUE
Mean : 101776.04874145007
Standard Deviation : 57385.775333702615
```

Answer:

Number of Observations : 5960

Frequency distributions of BAD :
0   4771
1   1189
Missing Values : 0

DEBTINC
Mean : 33.779915348721126
Standard Deviation : 8.60174618632853

LOAN
Mean : 18607.96979865772
Standard Deviation : 11207.480416694003

MORTDUE
Mean : 73760.817199559
Standard Deviation : 44457.60945841593

VALUE
Mean : 101776.04874145007
Standard Deviation : 57385.775333702615

b. (The Python code for this is in the file: 2_b.py)

Output Screenshot:

```
Number of Observations in the Train Partition : 4172
Number of Observations in the Test Partition : 1788

Frequency distributions of BAD in Train partition :
0    3344
1     828
Name: BAD, dtype: int64
Frequency distributions of BAD in Test partition :
0    1427
1     361
Name: BAD, dtype: int64

Mean and Standard Deviations in the Train Data
DEBTINC
Mean : 33.768030140847834
Standard Deviation : 8.444984923058202
LOAN
Mean : 18609.419942473633
Standard Deviation : 11300.340841755045
MORTDUE
Mean : 74067.9984938011
Standard Deviation : 44640.12848990737
VALUE
Mean : 101716.90242766062
Standard Deviation : 56671.2579097157

Mean and Standard Deviations in the Test Data
DEBTINC
Mean : 33.80787105552593
Standard Deviation : 8.96269755287088
LOAN
Mean : 18604.586129753916
Standard Deviation : 10990.880953150263
MORTDUE
Mean : 73055.47238643246
Standard Deviation : 44040.991080194915
VALUE
Mean : 101912.31917514124
Standard Deviation : 59015.12271676958
```

Answer:

Number of Observations in the Train Partition : 4172
Number of Observations in the Test Partition : 1788

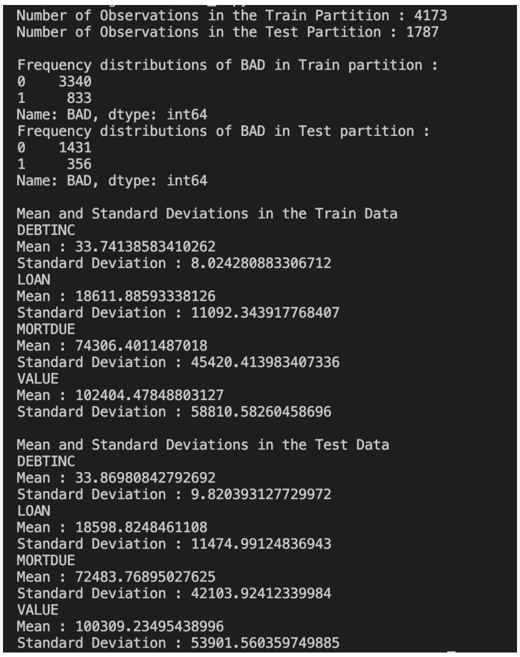
Frequency distributions of BAD in Train partition :
0 3344
1 828
Name: BAD, dtype: int64
Frequency distributions of BAD in Test partition :
0 1427
1 361
Name: BAD, dtype: int64

Mean and Standard Deviations in the Train Data
DEBTINC
Mean : 33.768030140847834
Standard Deviation : 8.444984923058202
LOAN
Mean : 18609.419942473633
Standard Deviation : 11300.340841755045
MORTDUE
Mean : 74067.9984938011
Standard Deviation : 44640.12848990737
VALUE
Mean : 101716.90242766062
Standard Deviation : 56671.2579097157

Mean and Standard Deviations in the Test Data
DEBTINC
Mean : 33.80787105552593
Standard Deviation : 8.96269755287088
LOAN
Mean : 18604.586129753916
Standard Deviation : 10990.880953150263
MORTDUE
Mean : 73055.47238643246
Standard Deviation : 44040.991080194915
VALUE
Mean : 101912.31917514124
Standard Deviation : 59015.12271676958

c. (The Python code for this is in the file: 2_c.py)

Output Screenshot:



```
Number of Observations in the Train Partition : 4173
Number of Observations in the Test Partition : 1787

Frequency distributions of BAD in Train partition :
0    3340
1     833
Name: BAD, dtype: int64
Frequency distributions of BAD in Test partition :
0    1431
1     356
Name: BAD, dtype: int64

Mean and Standard Deviations in the Train Data
DEBTINC
Mean : 33.74138583410262
Standard Deviation : 8.024280883306712
LOAN
Mean : 18611.88593338126
Standard Deviation : 11092.343917768407
MORTDUE
Mean : 74306.4011487018
Standard Deviation : 45420.413983407336
VALUE
Mean : 102404.47848803127
Standard Deviation : 58810.58260458696

Mean and Standard Deviations in the Test Data
DEBTINC
Mean : 33.86980842792692
Standard Deviation : 9.820393127729972
LOAN
Mean : 18598.8248461108
Standard Deviation : 11474.99124836943
MORTDUE
Mean : 72483.76895027625
Standard Deviation : 42103.92412339984
VALUE
Mean : 100309.23495438996
Standard Deviation : 53901.560359749885
```

Answer:

Number of Observations in the Train Partition : 4173
Number of Observations in the Test Partition : 1787

Frequency distributions of BAD in Train partition :
0   3340
1   833
Name: BAD, dtype: int64
Frequency distributions of BAD in Test partition :
0   1431
1   356
Name: BAD, dtype: int64

Mean and Standard Deviations in the Train Data
DEBTINC
Mean : 33.74138583410262
Standard Deviation : 8.024280883306712
LOAN
Mean : 18611.88593338126
Standard Deviation : 11092.343917768407
MORTDUE
Mean : 74306.4011487018
Standard Deviation : 45420.413983407336
VALUE
Mean : 102404.47848803127
Standard Deviation : 58810.58260458696

Mean and Standard Deviations in the Test Data
DEBTINC
Mean : 33.86980842792692
Standard Deviation : 9.820393127729972
LOAN
Mean : 18598.8248461108
Standard Deviation : 11474.99124836943
MORTDUE
Mean : 72483.76895027625
Standard Deviation : 42103.92412339984
VALUE
Mean : 100309.23495438996
Standard Deviation : 53901.560359749885

Question 3 Answers:

a. (The Python code for this is in the file: 3_a.py)

Output Screenshot :

```
Percent of investigations are found to be frauds : 19.9497
```

Answer:

Percent of investigations are found to be frauds : 19.9497

b. (The Python code for this is in the file: 3_b.py)

Output Screenshot :

```
Number of Observations in the Train Partition : 4768
Number of Observations in the Test Partition : 1192
```

Answer:

Number of Observations in the Train Partition : 4768
Number of Observations in the Test Partition : 1192

c. (The Python code for this is in the file: 3_c.py)

Output Screenshot :

```
k  MCE_Train  MCE_Test
2   0.124581  0.320470
3   0.230495  0.383389
4   0.313968  0.422819
5   0.387374  0.491611
6   0.225042  0.308725
7   0.259438  0.342282
```

Answer:

k	MCE_Train	MCE_Test
2	0.12458053691275167	0.32046979865771813
3	0.23049496644295303	0.38338926174496646
4	0.3139681208053691	0.4228187919463087
5	0.3873741610738255	0.49161073825503354
6	0.22504194630872484	0.3087248322147651
7	0.25943791946308725	0.3422818791946309

d. Answer :

From the above misclassification rates table of test partition, K=6 neighbours will yield the lowest misclassification rate in the Testing partition.

e. (The Python code for this is in the file: 3_e.py)

Output Screenshot :

```
Nearest Neighbours :
      TOTAL_SPEND  DOCTOR_VISITS  NUM_CLAIMS  MEMBER_DURATION  OPTOM_PRESC  NUM_MEMBERS
2967        16300              2           0              193            0            2
2980        16300              1           0              162            3            1
2962        16300             12           5              125            1            1
2976        16300              8           0              247            1            2
2977        16300              9           0              251            0            3
2969        16300              9           0              256            1            3

Fraud Probability Prediction :
It Might Not Be a Fraud
```

Answer:

Nearest Neighbours :

TOTAL_SPEND	DOCTOR_VISITS	NUM_CLAIMS	MEMBER_DURATION	OPTOM_PRESC	NUM_MEMBERS
16300	2	0	193	0	2
16300	1	0	162	3	1
16300	12	5	125	1	1
16300	8	0	247	1	2
16300	9	0	251	0	3
16300	9	0	256	1	3

Fraud Probability Prediction :
It Might Not Be a Fraud