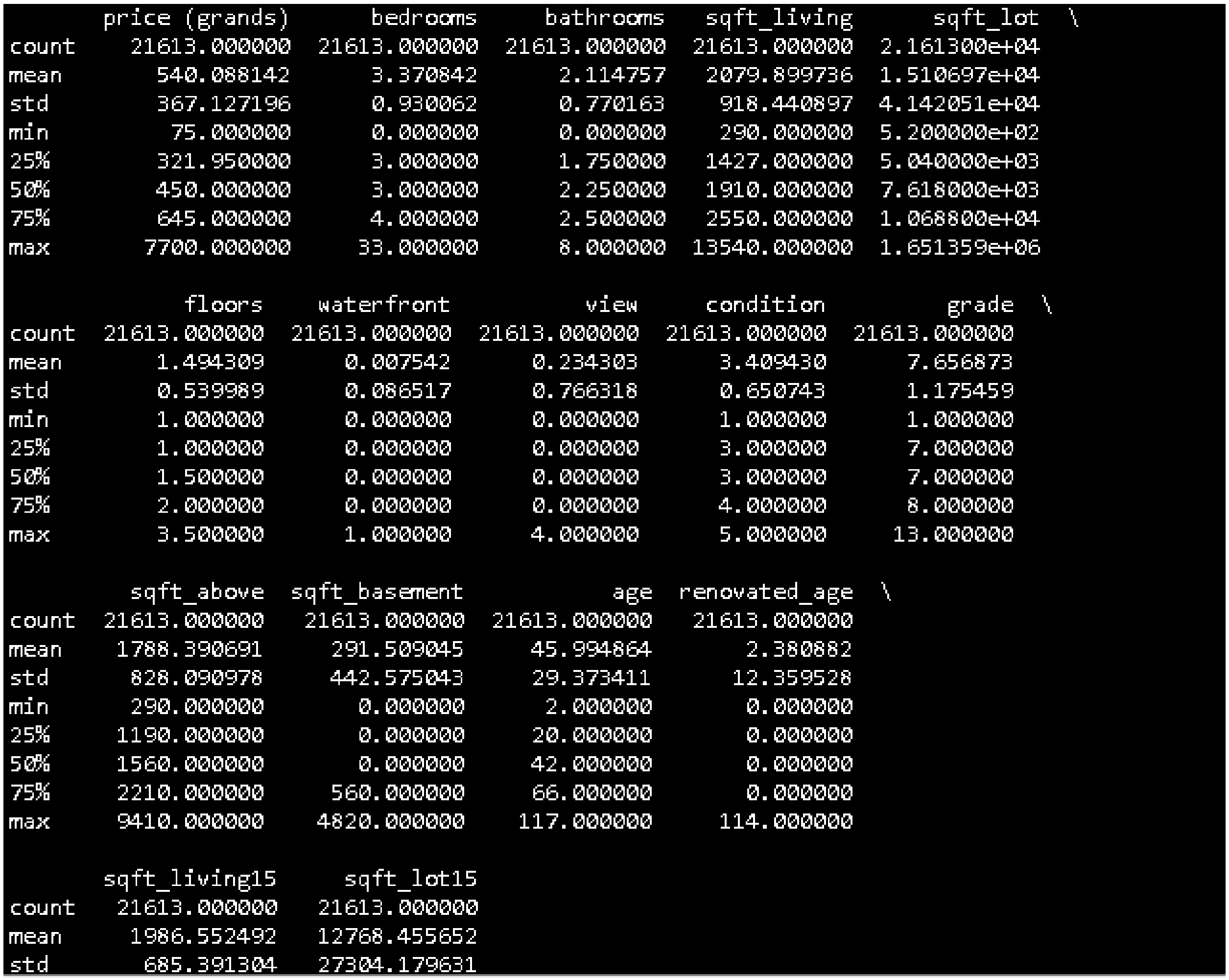
Solutions and Screenshots

A1

1.1

A data frame is a labeled two-dimensional data structure containing columns that can be of different types.



```
       price (grands)      bedrooms     bathrooms   sqft_living      sqft_lot  \
count    21613.000000  21613.000000  21613.000000  21613.000000  2.161300e+04
mean       540.088142      3.370842      2.114757   2079.899736  1.510697e+04
std        367.127196      0.930062      0.770163    918.440897  4.142051e+04
min         75.000000      0.000000      0.000000    290.000000  5.200000e+02
25%        321.950000      3.000000      1.750000   1427.000000  5.040000e+03
50%        450.000000      3.000000      2.250000   1910.000000  7.618000e+03
75%        645.000000      4.000000      2.500000   2550.000000  1.068800e+04
max       7700.000000     33.000000      8.000000  13540.000000  1.651359e+06

             floors    waterfront          view     condition         grade  \
count  21613.000000  21613.000000  21613.000000  21613.000000  21613.000000
mean       1.494309      0.007542      0.234303      3.409430      7.656873
std        0.539989      0.086517      0.766318      0.650743      1.175459
min        1.000000      0.000000      0.000000      1.000000      1.000000
25%        1.000000      0.000000      0.000000      3.000000      7.000000
50%        1.500000      0.000000      0.000000      3.000000      7.000000
75%        2.000000      0.000000      0.000000      4.000000      8.000000
max        3.500000      1.000000      4.000000      5.000000     13.000000

         sqft_above  sqft_basement           age  renovated_age  \
count  21613.000000   21613.000000  21613.000000   21613.000000
mean    1788.390691     291.509045     45.994864       2.380882
std      828.090978     442.575043     29.373411      12.359528
min      290.000000       0.000000      2.000000       0.000000
25%     1190.000000       0.000000     20.000000       0.000000
50%     1560.000000       0.000000     42.000000       0.000000
75%     2210.000000     560.000000     66.000000       0.000000
max     9410.000000    4820.000000    117.000000     114.000000

       sqft_living15     sqft_lot15
count   21613.000000   21613.000000
mean     1986.552492   12768.455652
std       685.391304   27304.179631
```

1.2 In python file

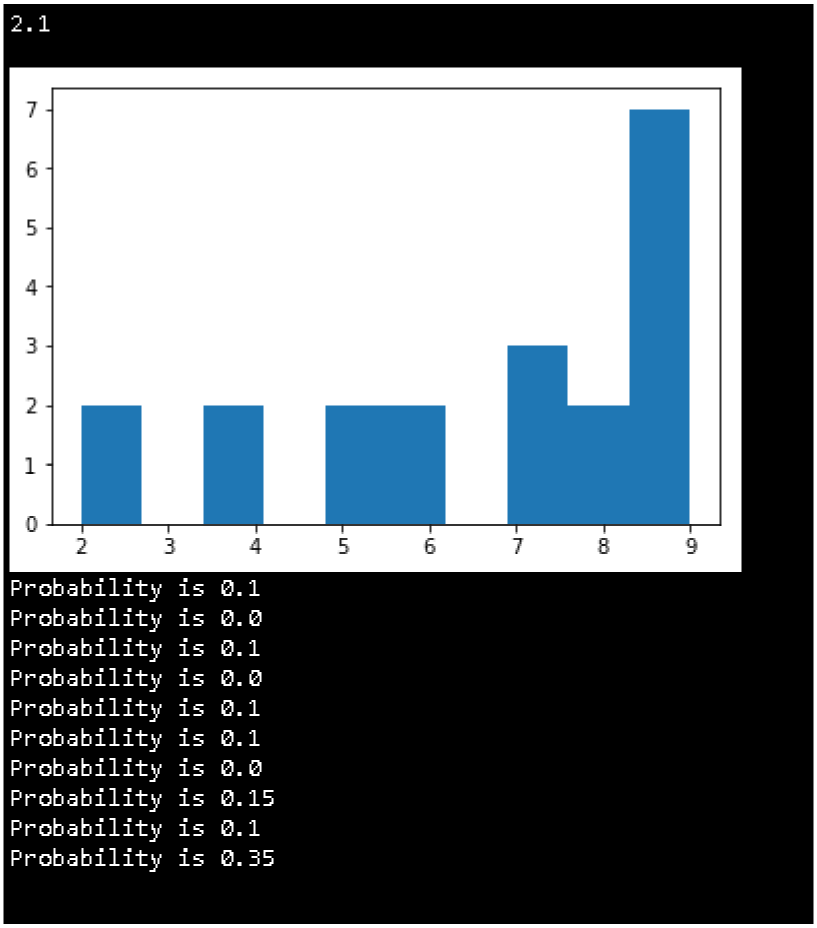
1.3 Best model obtained is with using learning rate 1.0. This wouldn't create a linear line as N dimensional is projected on to 2 d plane.

1.4 In Sol1 python file

1.5 In sol1 python file

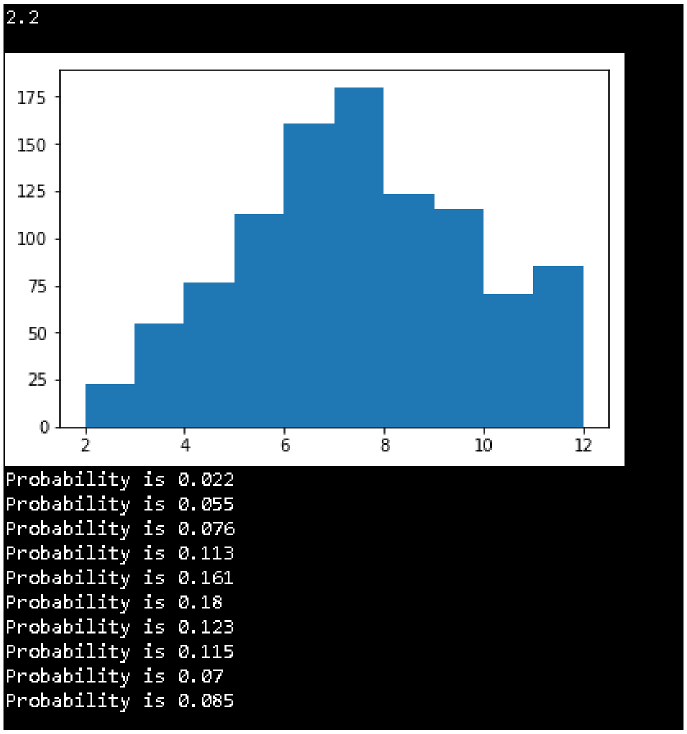
A2

2.1



```
Probability is 0.1
Probability is 0.0
Probability is 0.1
Probability is 0.0
Probability is 0.1
Probability is 0.1
Probability is 0.0
Probability is 0.15
Probability is 0.1
Probability is 0.35
```

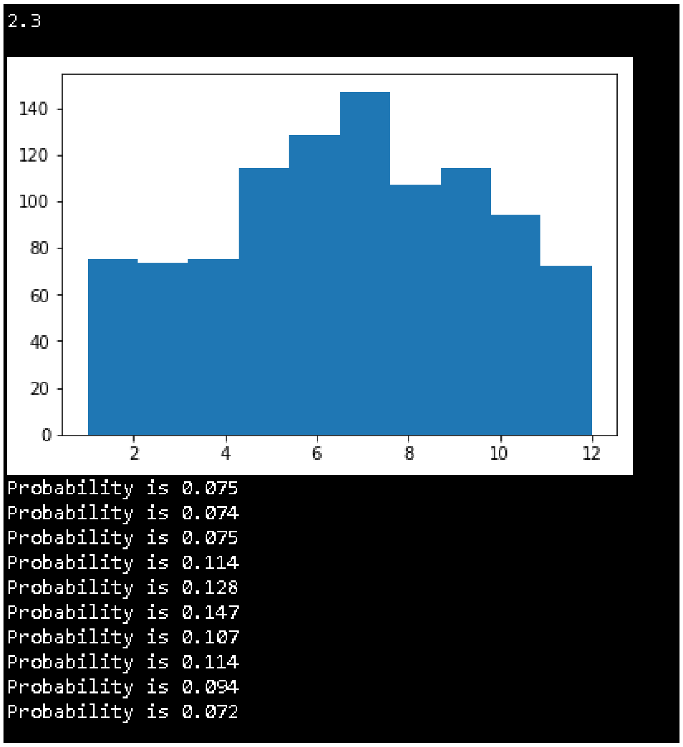
2.2



```
2.2
```

```
Probability is 0.022
Probability is 0.055
Probability is 0.076
Probability is 0.113
Probability is 0.161
Probability is 0.18
Probability is 0.123
Probability is 0.115
Probability is 0.07
Probability is 0.085
```

2.3



```
2.3
```

```
Probability is 0.075
Probability is 0.074
Probability is 0.075
Probability is 0.114
Probability is 0.128
Probability is 0.147
Probability is 0.107
Probability is 0.114
Probability is 0.094
Probability is 0.072
```

A3

Probability of sensor to be faulty P(faulty) = 0.01

Probability of sensor to be not faulty P (¬ faulty) = 0.99

Probability of getting a reading less than 1 when sensor is faulty = P(<1|faulty) = 1.0

Probability of getting a reading less than 1 when sensor is not faulty = P (<1|¬ faulty) = .33

$$P(faulty|<1) = \frac{P(<1|faulty) *P(faulty)}{P(<1|faulty) *P(faulty) + P (<1|\neg faulty) * P (\neg faulty)}$$

For 1st reading by the sensor = 1*.01/ (1*.01 + 1/3*.99) = .029

For 2nd consecutive reading by the sensor = 1*.01/ (1*.01 + 1/3*1/3*.99) = .083

Similarly, for 10th reading = 1*.01/ (1*.01 + 1/3^10*.99) = .998

So, model will be = 1*.01/ (1*.01 + 1/3^N*.99)

A4

The mean (µ1) of the first class is 3 and second (µ2) is 1. The Bell curve will intersect at 2. The standard deviation is 1. The Standard Score (z) is calculated by

$$z = (x - \mu1)/ \sigma$$

So, z will be -1. The probability of area under the curve onto the left is .1587. So, probability of remaining area is 1 - .1587 = 0.8413. That is why the theoretical limit will be around 84%.

Another way to prove that is 68/95/99.7 rule. The first 68% area under the bell curve lies at the first standard deviation from the mean which is 0 and 2. Rest we must find the area between -1 to 0. As bell curves are symmetrical, remaining area will be half of 32% which is 16%. So, theoretical limit will be 68% + 16% = 84%

References

[1] "Intro to Data Structures — pandas 0.20.3 documentation", Pandas.pydata.org, 2017. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/dsintro.html. [Accessed: 03- Oct- 2017].

[2] "numpy.random.choice — NumPy v1.13 Manual", Docs.scipy.org, 2017. [Online]. Available: https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.random.choice.html. [Accessed: 03- Oct- 2017].

[3] F. Y-scaling?, "Fitting a Gaussian to a histogram with MatPlotLib and Numpy - wrong Y-scaling?", Stackoverflow.com, 2017. [Online]. Available: https://stackoverflow.com/questions/23447262/fitting-a-gaussian-to-a-histogram-with-matplotlib-and-numpy-wrong-y-scaling. [Accessed: 03- Oct- 2017].

[4] 2017. [Online]. Available: http://www.stat.ufl.edu/~athienit/Tables/Ztable.pdf. [Accessed: 03- Oct- 2017].