STA HOME WORK-1

Question-1:

These are the results obtained using different methods (cosine and jaccard) for different vectors.

a)

Jaccard similarity

```
> distance (B, method="jaccard")

    v1    v2    v3
v1 0.0   0.5    0.5
v2 0.5   0.0    0.5
v3 0.5   0.5    0.0
```

Cosine similarity.

```
>distance (B, method="cosine")
          v1        v2        v3
v1 1.0000000 0.6666667 0.6666667
v2 0.6666667 1.0000000 0.6666667
v3 0.6666667 0.6666667 1.0000000
```

b)

Jaccard similarity

```
> distance(B, method="jaccard")
          v1        v2        v3
v1 0.0000000 0.6000000 0.6666667
v2 0.6000000 0.0000000 0.8333333
v3 0.6666667 0.8333333 0.0000000
```

Cosine similarity

```
> distance(B, method="cosine")
          v1        v2        v3
v1 1.0000000 0.5773503 0.5000000
v2 0.5773503 1.0000000 0.2886751
v3 0.5000000 0.2886751 1.0000000
```

c)

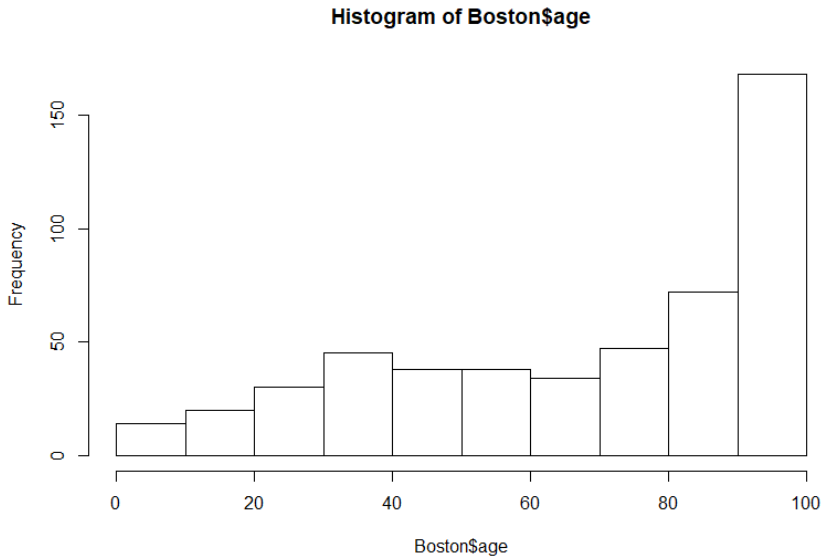
Cosine similarity

```
> distance(B, method="cosine")
          v1        v2        v3
v1 1.0000000 0.6604744 0.5989377
v2 0.6604744 1.0000000 0.5138701
v3 0.5989377 0.5138701 1.0000000
```

Question-2

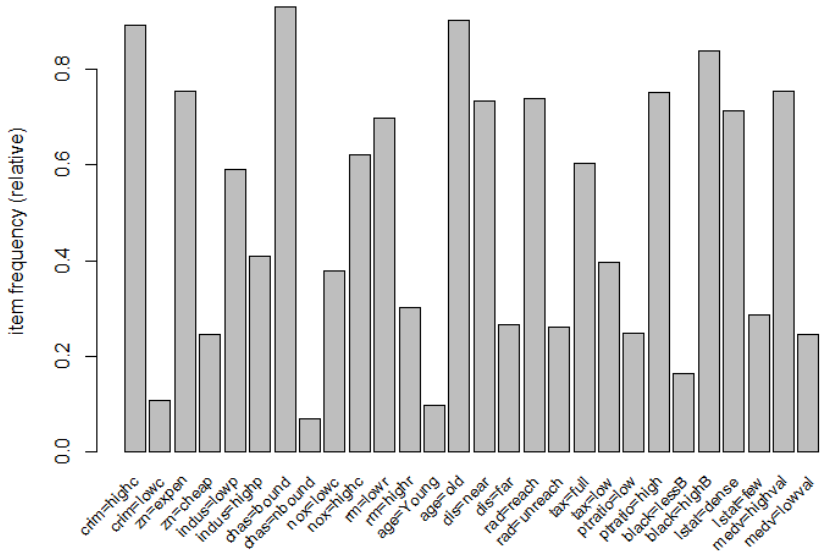
To transform the data into a binary incidence matrix, we have to understand the data by plotting the histograms of the data.

For example,



A binarised split of >25 and <25 is taken for age after looking at the graphs.

The frequency plot obtained:



C)

Low crime rules:

```
> inspect(head(sort(rulescrimelow, by = "confidence"), n = 3))
    lhs                                      rhs         support     confidence lift     count
[1] {black=lessB,lstat=few,medv=lowval}  => {crim=lowc} 0.001976285  1          9.37037  1
[2] {rm=highr,black=lessB,medv=lowval}   => {crim=lowc} 0.001976285  1          9.37037  1
[3] {chas=bound,black=lessB,medv=lowval} => {crim=lowc} 0.003952569  1          9.37037  2
```

Low Distance rules:

```
> inspect(head(sort(rulescrimehigh, by = "confidence"), n = 3))
     lhs                           rhs        support    confidence lift     count
[1] {chas=nbound,age=Young} => {dis=near} 0.00197628    1          3.748148  1
[2] {age=Young,ptratio=low} => {dis=near} 0.043478261   1          3.748148  22
[3] {age=Young,tax=low}     => {dis=near} 0.007905138   1          3.748148  4
```

d)

Low ptratio rules:

```
> inspect(head(sort(rulespupilratio,by="lift")),n=5)
   lhs                                       rhs            support     confidence lift   count
[1] {chas=nbound,age=Young}                => {ptratio=low}  0.001976285 1          4.048  1
[2] {chas=nbound,dis=far}                  => {ptratio=low}  0.003952569 1          4.048  2
[3] {zn=cheap,nox=highc}                   => {ptratio=low}  0.025691700 1          4.048  13
[4] {chas=nbound,age=Young,medv=lowval} => {ptratio=low}  0.001976285 1          4.048  1
[5] {zn=cheap,chas=nbound,age=Young}       => {ptratio=low}  0.001976285 1          4.048  1
[6] {chas=nbound,age=Young,dis=far}        => {ptratio=low}  0.001976285 1          4.048  1
```

Inference:

From the rules above it is better to opt for a place where,

the black proportion is less

the distance to employment center is less

property tax to be low and

median value of homes is more .

Question – 3

First install the required packages and load the appropriate libraries, then create a random data that is similar to the original form of data. And also a target variable with class as 1. Permute the features next and name the target as 0.

Build a decision tree model and after that we get a tree that has only one root indicating that classification can’t be done using the features.