192211833 DAY 4:

PROB 1:

INPUT:

@relation dataset

@attribute customerID{1,2,3,4,5}

@attribute gender{male,female}

@attribute Age{19,21,20,23,31}

@attribute Annual\_income{15,16,17}

@attribute spending\_score{39,81,6,77,40}

@data

1 male 19 15 39

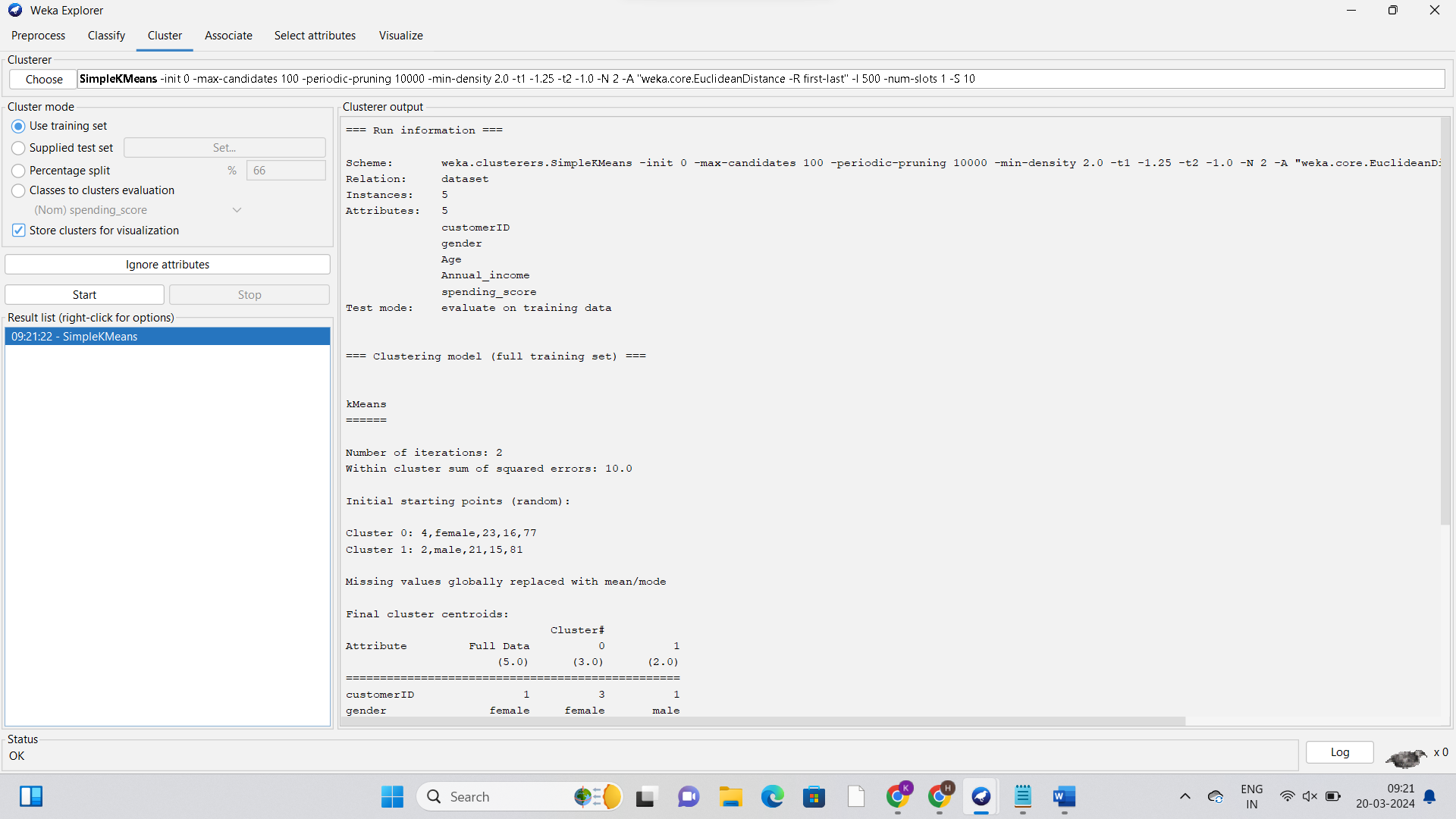
2 male 21 15 81

3 female 20 16 6

4 female 23 16 77

5 female 31 17 40

OUTPUT:



PROB2:

INPUT:

@relation dataset

@attribute employee{111,222,333,444,555,666}

@attribute gender{male,female}

@attribute Age{28,25,26,30,29}

@attribute Salary{150000,160000,170000,200000}

@attribute Credit{39,27,42,40,64,72}

@data

111 male 28 150000 39

222 male 25 150000 27

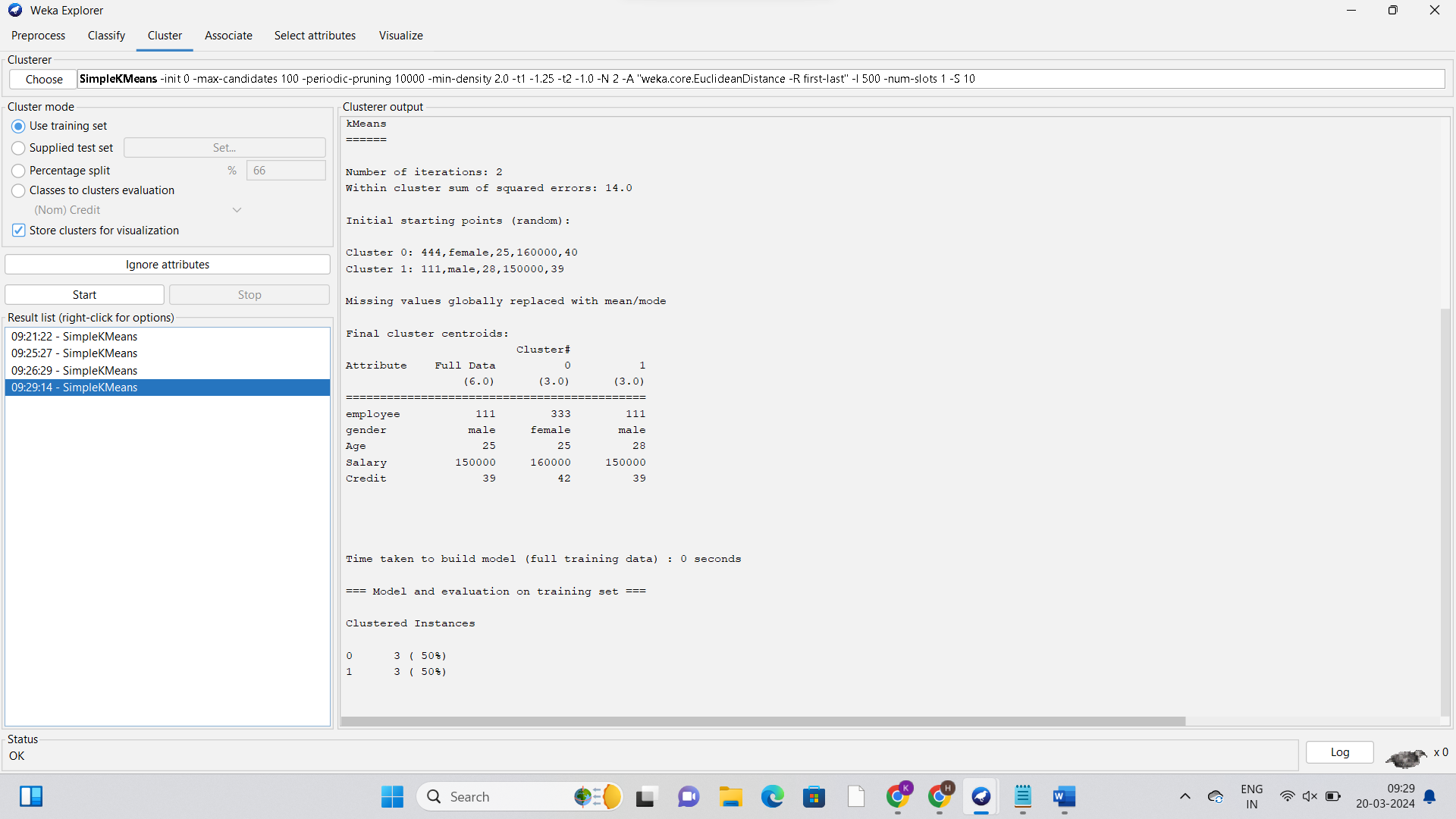
333 female 26 160000 42

444 female 25 160000 40

555 female 30 170000 64

666 male 29 200000 72

OUPUT:



PROB5:

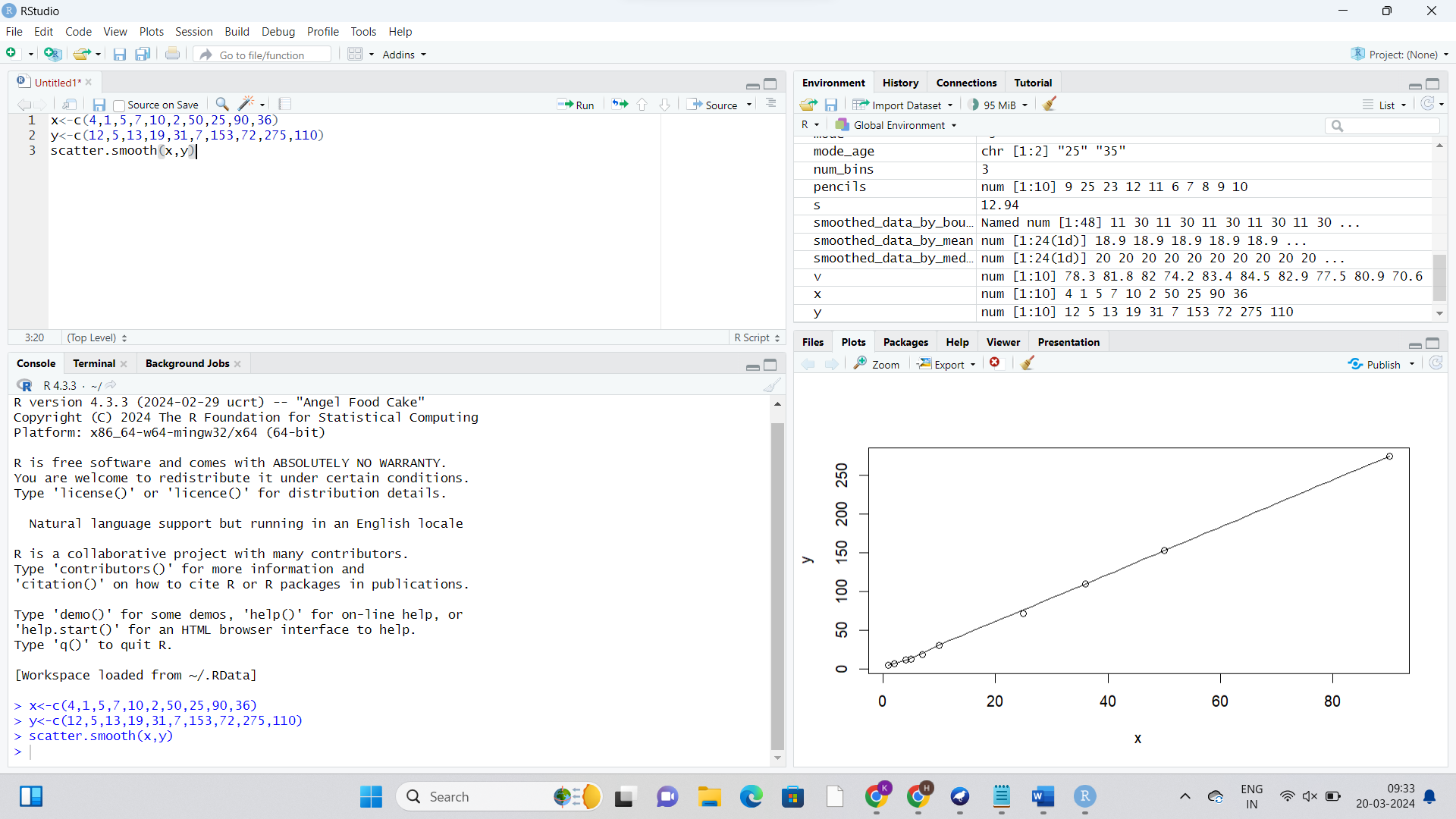
INPUT:

x<-c(4,1,5,7,10,2,50,25,90,36)

y<-c(12,5,13,19,31,7,153,72,275,110)

scatter.smooth(x,y)

OUTPUT:



PROB6:

INPUT:

@relation items

@attribute bread{true,false}

@attribute cheese{true,false}

@attribute egg{true,false}

@attribute juice{true,false}

@attribute milk{true,false}

@attribute yogurt{true,false}

@data

true true true true false false

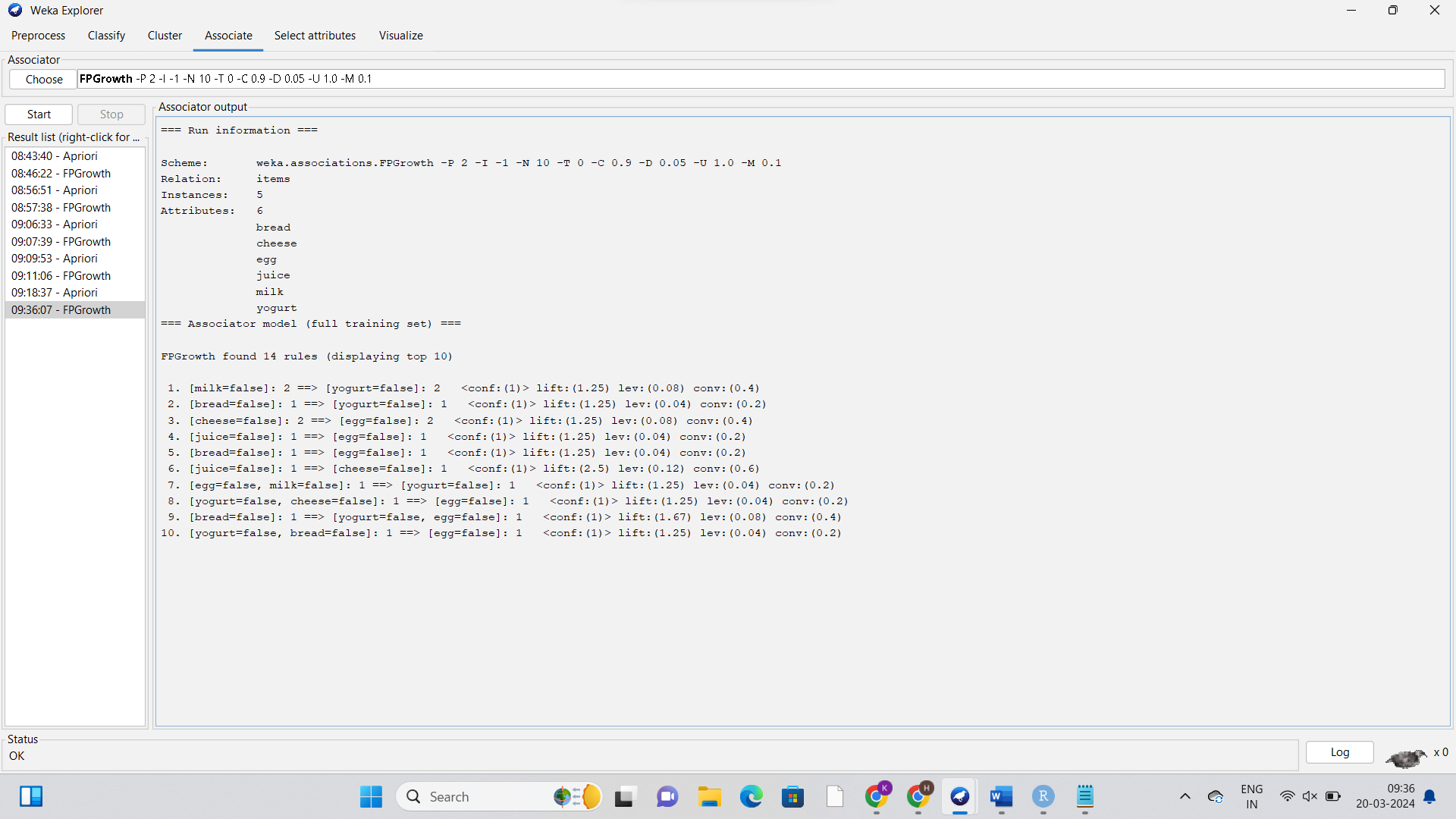
true true false true false false

true false false false true true

true false false true true false

false true false true true false

OUTPUT:



PROB8:

INPUT:

# Sales price records

sales <- c(5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215)

# (a) Equal-frequency (equi-depth) partitioning

eq\_freq\_bins <- split(sales, cut(sales, breaks = 3))

# (b) Equal-width partitioning

eq\_width\_bins <- split(sales, cut(sales, breaks = seq(min(sales), max(sales), length.out = 4)))

# (c) Clustering

num\_bins <- 3

clustering\_bins <- split(sales, kmeans(matrix(sales), centers = num\_bins)$cluster)

# Display the results

print("Equal-frequency (equi-depth) partitioning:")

print(eq\_freq\_bins)

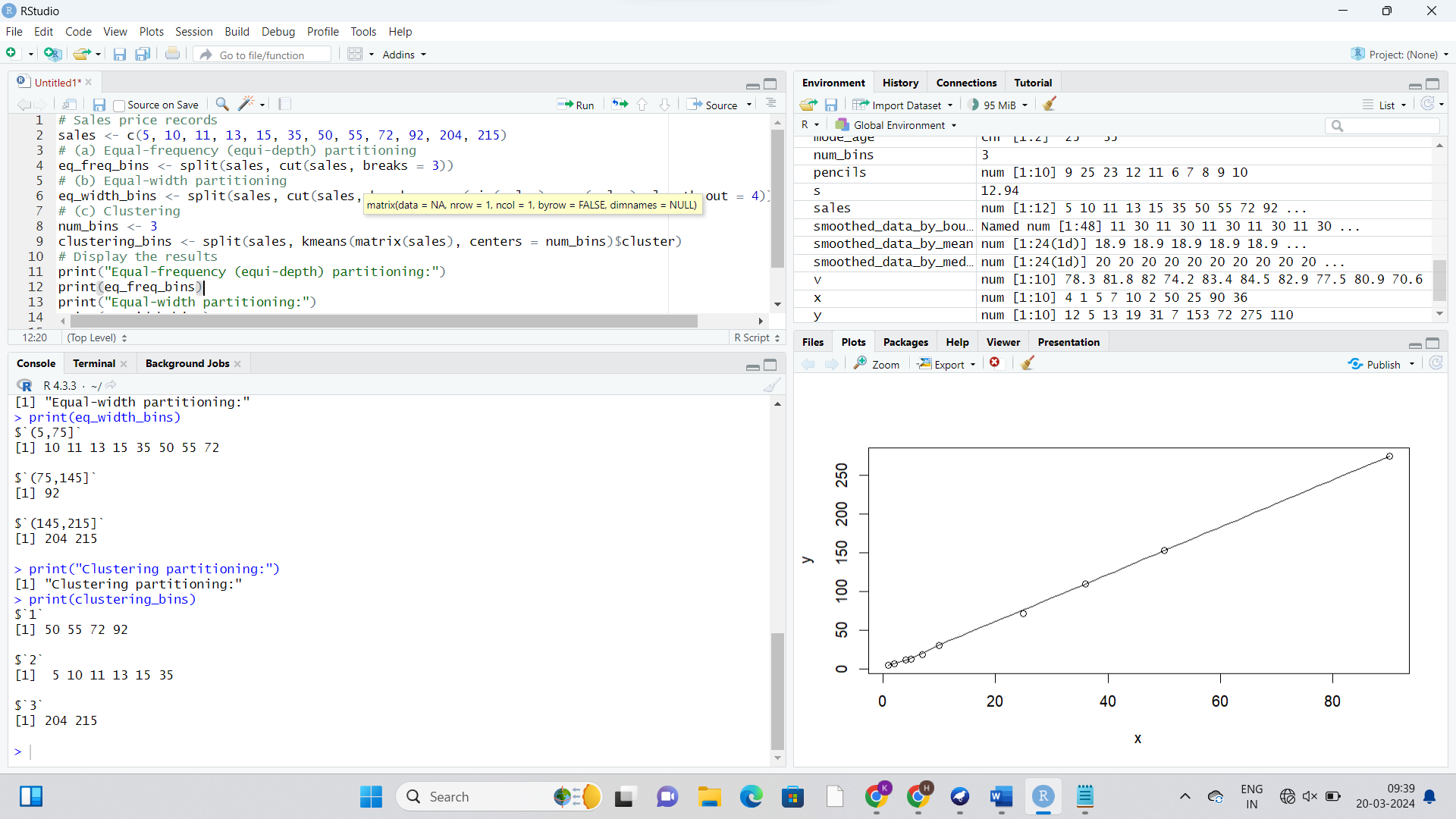
print("Equal-width partitioning:")

print(eq\_width\_bins)

print("Clustering partitioning:")

print(clustering\_bins)

OUTPUT:



PROB 10:

INPUT:

@relation dataset

@attribute sony{t,f}

@attribute bpl{t,f}

@attribute LG{t,f}

@attribute samsung{t,f}

@attribute onida{t,f}

@data

t t t f f

f t f t f

f t f f t

t t f t f

f t f f t

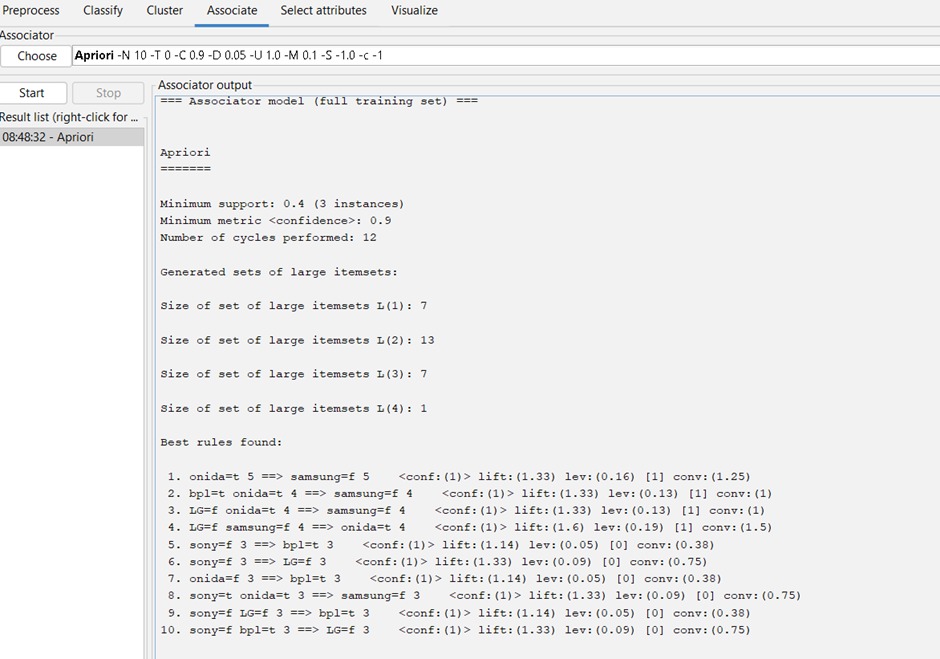
t f f f t

t t t f t

t t f f t

t,t,t,f,t

t,t,f,f,t



PROB11:

INPUT:

x<-c(100,70,60,90,90)

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

min\_max\_normalized<-min\_max\_normal(x)

print(min\_max\_normalized)

z\_score\_normal<-function(x){(x-mean(x))/sd(x)}

z\_score\_normalized<-z\_score\_normal(x)

print(z\_score\_normalized)

z\_score\_mad<-function(x){(x-mean(x))/mad(x)}

z\_score\_mad\_data<-z\_score\_mad(x)

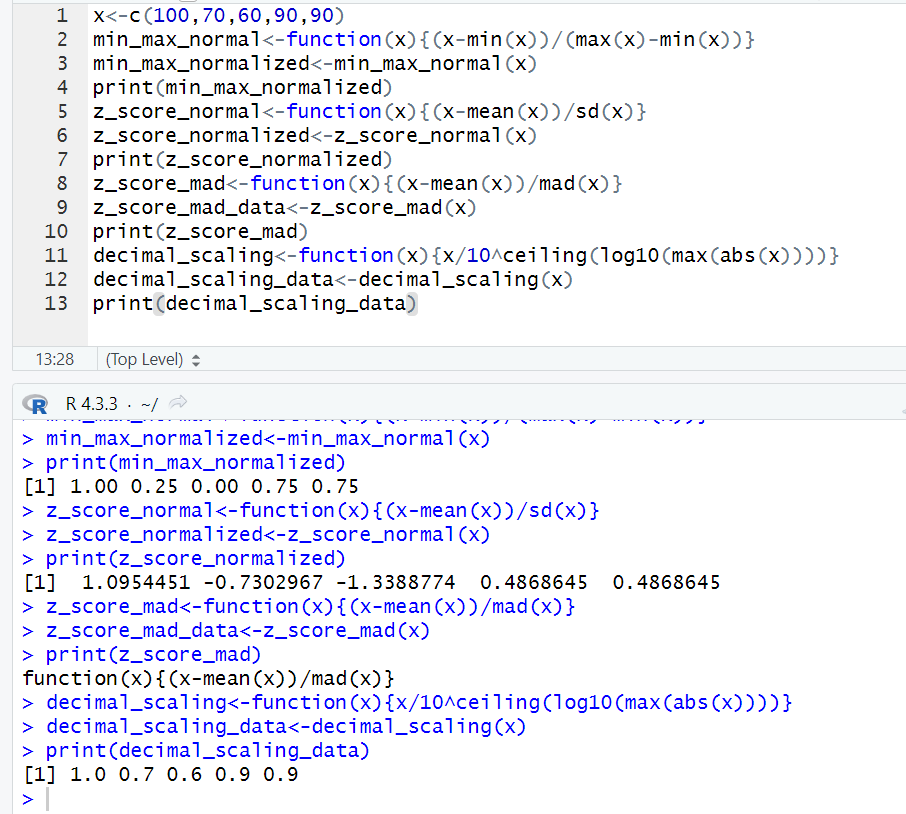
print(z\_score\_mad)

decimal\_scaling<-function(x){x/10^ceiling(log10(max(abs(x))))}

decimal\_scaling\_data<-decimal\_scaling(x)

print(decimal\_scaling\_data)

OUTPUT:



PROB 12:

INPUT:

avg\_speed<-c(78,81,82,74,83,82,77,80,70)

avg\_time<-c(39,37,36,42,35,36,40,38,46)

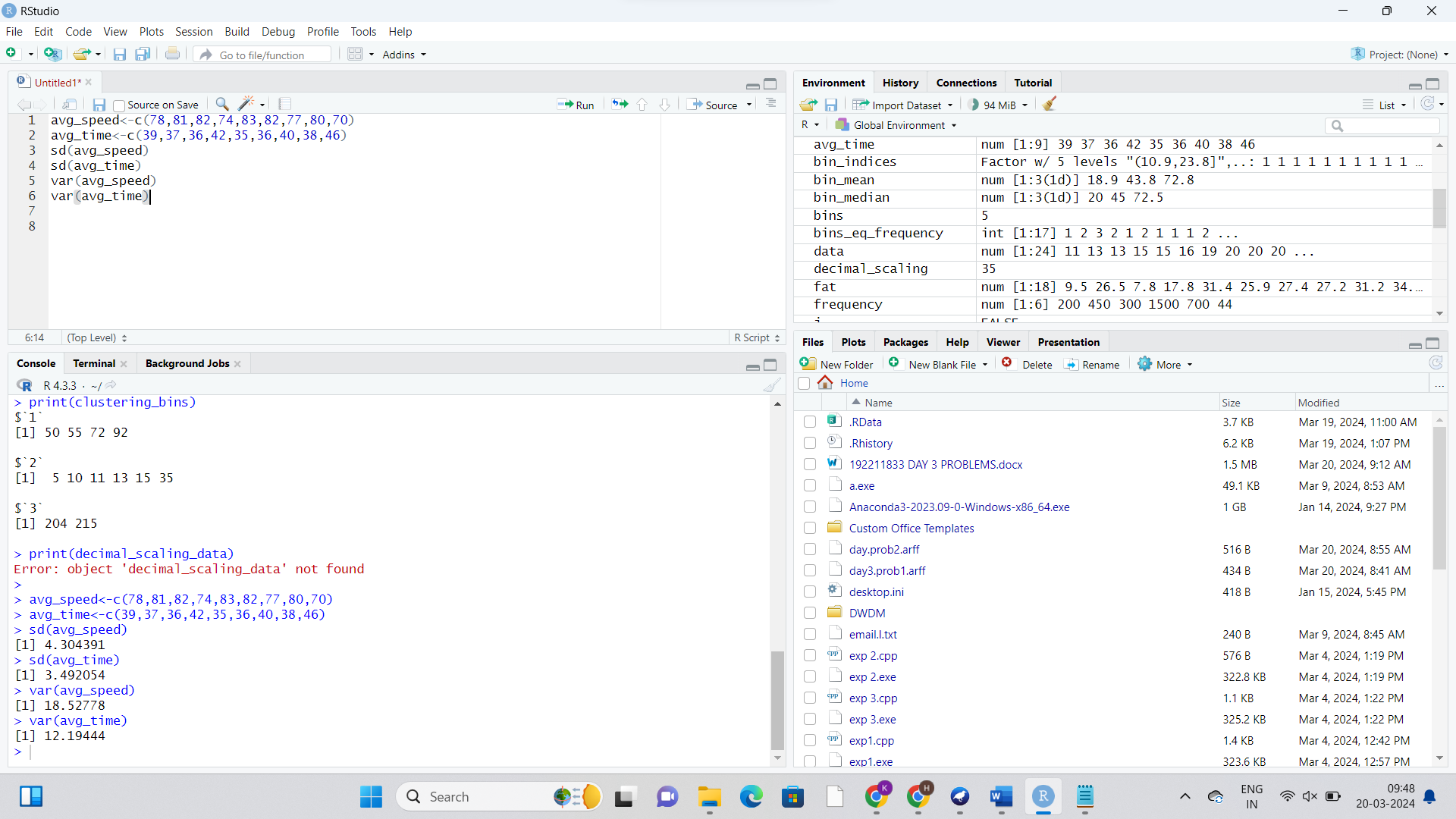
sd(avg\_speed)

sd(avg\_time)

var(avg\_speed)

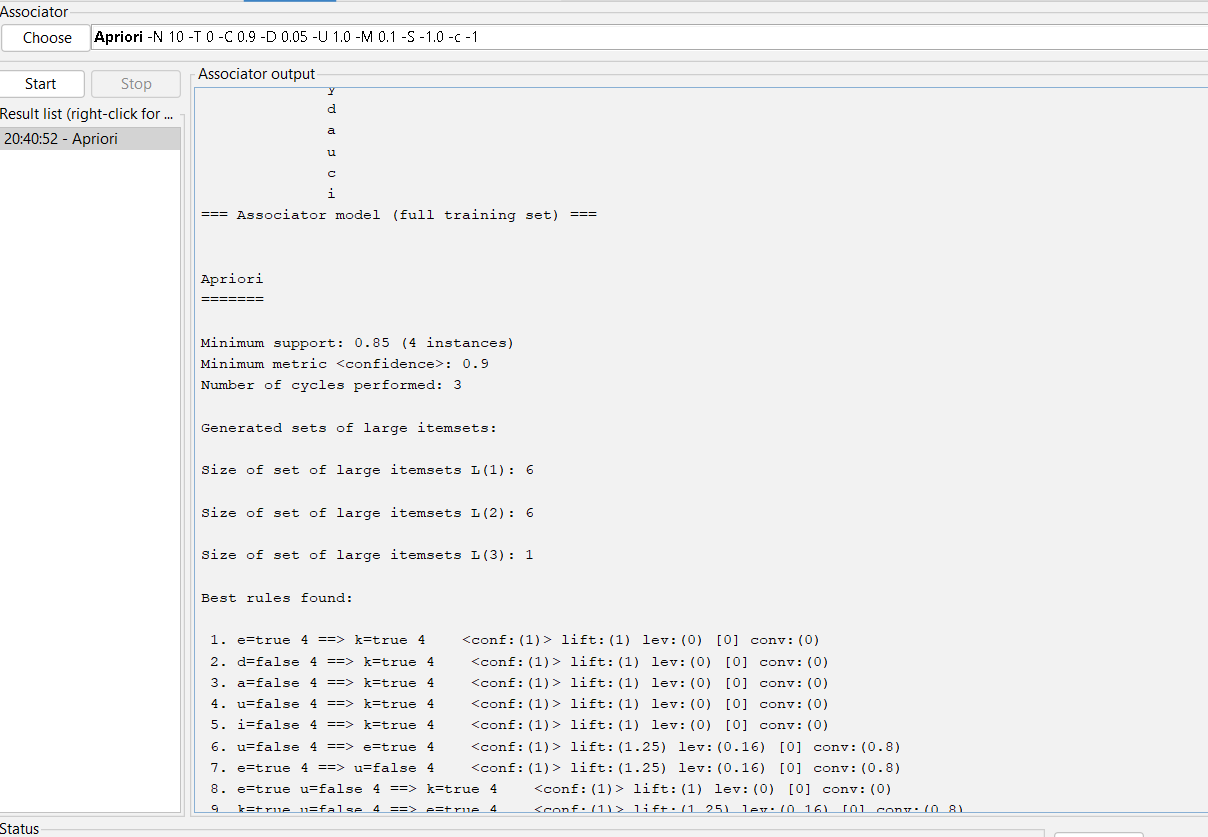
var(avg\_time)

OUTPUT:



PROB13:

APRIORI ALGORITHM:



FP GROWTH:

