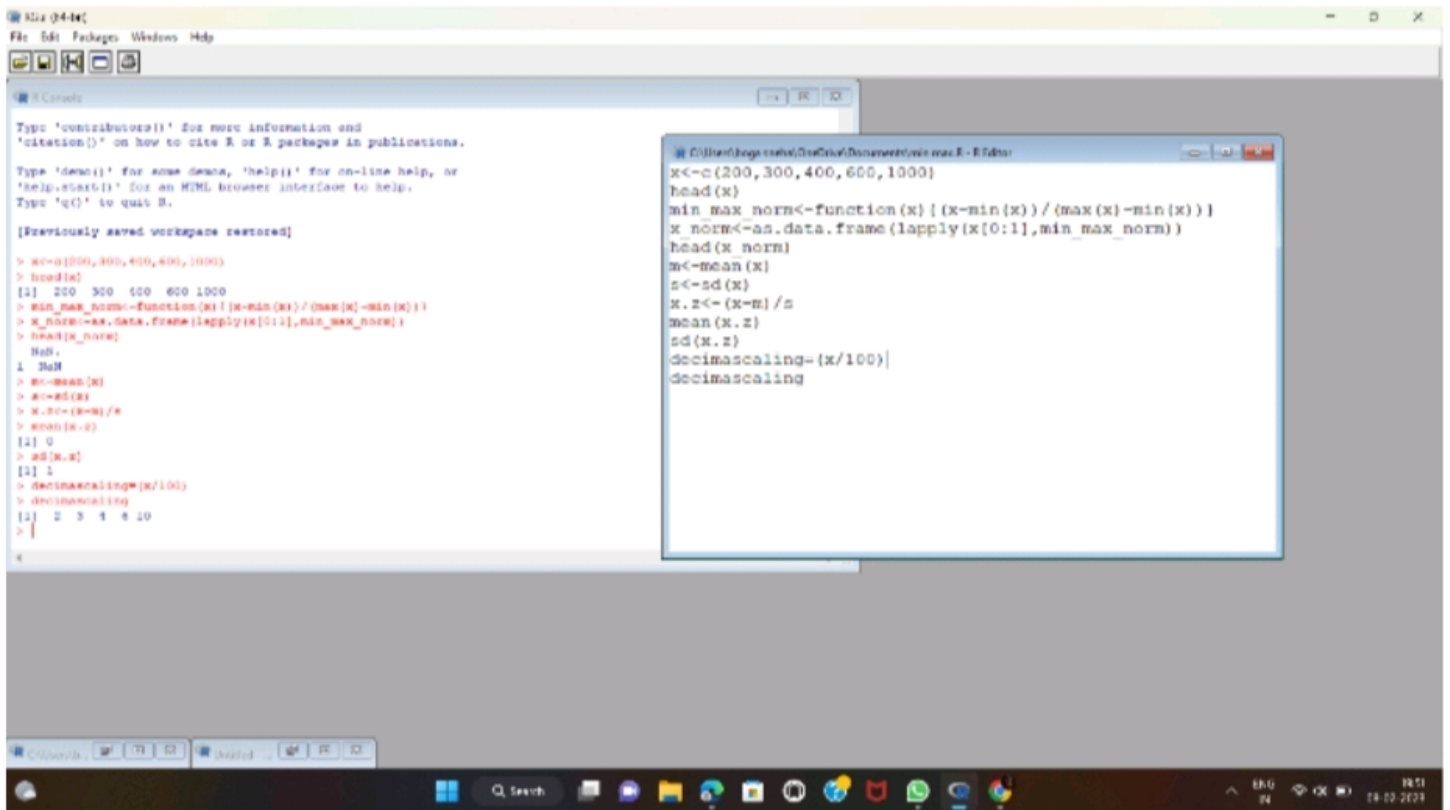


2)



The screenshot shows an RStudio interface with a script editor on the left and a console on the right. A small window titled 'R Editor' is also open, displaying the same code as the script editor. The code defines a vector 'x' with values 200, 300, 400, 600, 1000. It then defines a function 'min_max_norm' that takes a vector 'x' and returns a data frame with the original values and their normalized values. The code then applies this function to 'x' and stores the result in 'x_norm'. Finally, it calculates the mean and standard deviation of 'x' and 'x_norm', and performs a decimal scaling operation on 'x'.

```
x<-c(200,300,400,600,1000)
head(x)
min_max_norm<-function(x){(x-min(x))/(max(x)-min(x))}
x_norm<-as.data.frame(lapply(x[0:1],min_max_norm))
head(x_norm)
m<-mean(x)
s<-sd(x)
x.z<-(x-m)/s
mean(x.z)
sd(x.z)
decimalscaling=(x/100)
decimalscaling
```

```
[Previously saved workspace restored]

> x<-c(200,300,400,600,1000)
> head(x)
[1] 200 300 400 600 1000
> min_max_norm<-function(x){(x-min(x))/(max(x)-min(x))}
> x_norm<-as.data.frame(lapply(x[0:1],min_max_norm))
> head(x_norm)
  x_norm
1 200
2 300
3 400
4 600
5 1000
> m<-mean(x)
[1] 400
> s<-sd(x)
[1] 360.5551
> x.z<-(x-m)/s
[1] 0.5547002 0.8320503 1.1184515 1.6651514 2.7735009
> mean(x.z)
[1] 1
> sd(x.z)
[1] 1
> decimalscaling=(x/100)
> decimalscaling
[1] 2 3 4 6 10
```