```
> #import train data
>  train_data <-  read.csv("blogData_train.csv",1)
> #a. Read the dataset and identify the right features.
>  #Columns 51,52,55,56,57,60,61,62,263 to 269, 270 to 276, 277, 278, 279,280
, 281
>  #have been identified pdf attached describing attributes.
>  train_data1<- train_data[c(51,52,55,56,57,60,61,62,263:269, 270:276, 277,
278, 279,280, 281)]
> #Clean dataset, impute missing values and perform exploratory data
>  #analysis.
>  #Visualising missing values with naniar package
>  library(naniar)
> #c. Visualize the dataset and make inferences from that.
> colnames(train_data1) <- seq(1:27)
> head(train_data1)
  1 2  3 4 5 6  7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
1 6 2 -2 0 0 0 35 0 0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0
2 6 2 -2 0 0 0 35 0 0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0
3 2 2  2 0 0 0 10 0 0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0
4 3 1 -1 0 0 0 34 0 0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0
5 6 0 -2 0 0 0 59 0 0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0
6 6 0 -2 0 0 0 59 0 0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0
  27
1  0
2  0
3  1
4 27
5  0
6  0
> train_data1 <- as.data.frame(train_data1)
> str(train_data1)
'data.frame':  52396 obs. of  27 variables:
 $ 1 : num  6 6 2 3 6 6 3 30 30 0 ...
 $ 2 : num  2 2 2 1 0 0 1 27 27 0 ...
 $ 3 : num  -2 -2 2 -1 -2 -2 -1 26 26 0 ...
 $ 4 : num  0 0 0 0 0 0 0 0 0 2 ...
 $ 5 : num  0 0 0 0 0 0 0 0 0 2 ...
 $ 6 : num  0 0 0 0 0 0 0 0 0 2 ...
 $ 7 : num  35 35 10 34 59 59 34 58 58 11 ...
 $ 8 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ 9 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ 10: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 11: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 12: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 13: num  1 1 1 0 0 0 0 0 0 0 ...
 $ 14: num  0 0 0 1 1 1 1 0 0 0 ...
 $ 15: num  0 0 0 0 0 0 0 1 1 1 ...
 $ 16: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 17: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 18: num  1 1 0 0 1 1 0 0 0 0 ...
 $ 19: num  0 0 1 1 0 0 1 1 1 0 ...
 $ 20: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 21: num  0 0 0 0 0 0 0 0 0 1 ...
 $ 22: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 23: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 24: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 25: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 26: num  0 0 0 0 0 0 0 0 0 0 ...
 $ 27: num  0 0 1 27 0 0 27 9 9 0 ...


> cor.test(train_data1[,2],train_data1[,3])
```

```
        Pearson's product-moment correlation

data:  train_data1[, 2] and train_data1[, 3]
t = 180.02, df = 52394, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6128659 0.6234469
sample estimates:
      cor
0.6181844

> cor.test(train_data1[,4],train_data1[,5])

        Pearson's product-moment correlation

data:  train_data1[, 4] and train_data1[, 5]
t = 181.03, df = 52394, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6150203 0.6255560
sample estimates:
      cor
0.6203161

> cor.test(train_data1[,3],train_data1[,4])

        Pearson's product-moment correlation

data:  train_data1[, 3] and train_data1[, 4]
t = -3.5014, df = 52394, p-value = 0.0004632
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.023854349 -0.006733344
sample estimates:
       cor
-0.01529497

>
```