Q1) Write WordCount program using Hadoop

CODE:

pom.xml

```
<dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-common</artifactId>
        <version>3.3.3</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-mapreduce-client-core</artifactId>
        <version>3.3.3</version>
    </dependency>
```

WC_Mapper.java

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class WC_Mapper extends MapReduceBase implements
Mapper<LongWritable,Text,Text,IntWritable>{
   private final static IntWritable one = new IntWritable(1);
   private Text word = new Text();
   public void map(LongWritable key, Text value,OutputCollector<Text,IntWritable> output,
                Reporter reporter) throws IOException{
      String line = value.toString();
      StringTokenizer  tokenizer = new StringTokenizer(line);
      while (tokenizer.hasMoreTokens()){
         word.set(tokenizer.nextToken());
         output.collect(word, one);
      }
   }

}
```

WC_Reducer

```
import java.io.IOException;
import java.util.Iterator;
```

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WC_Reducer  extends MapReduceBase implements
Reducer<Text,IntWritable,Text,IntWritable> {
   public void reduce(Text key, Iterator<IntWritable> values,OutputCollector<Text,IntWritable>
output,
                   Reporter reporter) throws IOException {
      int sum=0;
      while (values.hasNext()) {
         sum+=values.next().get();
      }
      output.collect(key,new IntWritable(sum));
   }
}
```

WC_Runner

```
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.TextOutputFormat;
public class WC_Runner {
   public static void main(String[] args) throws IOException{
      JobConf conf = new JobConf(WC_Runner.class);
      conf.setJobName("WordCount");
      conf.setOutputKeyClass(Text.class);
      conf.setOutputValueClass(IntWritable.class);
      conf.setMapperClass(WC_Mapper.class);
      conf.setCombinerClass(WC_Reducer.class);
      conf.setReducerClass(WC_Reducer.class);
      conf.setInputFormat(TextInputFormat.class);
      conf.setOutputFormat(TextOutputFormat.class);
      FileInputFormat.setInputPaths(conf,new Path(args[0]));
      FileOutputFormat.setOutputPath(conf,new Path(args[1]));
```

```
        JobClient.runJob(conf);
    }
}
```

Input file

File contents

```
hi
my name is jagruti
this is so cool
```

Output file

File contents

```
cool 1
hi    1
is    2
jagruti    1
my   1
name    1
so    1
this  1
```

Q2)Using Power Pivot (Excel) Perform the following on any dataset

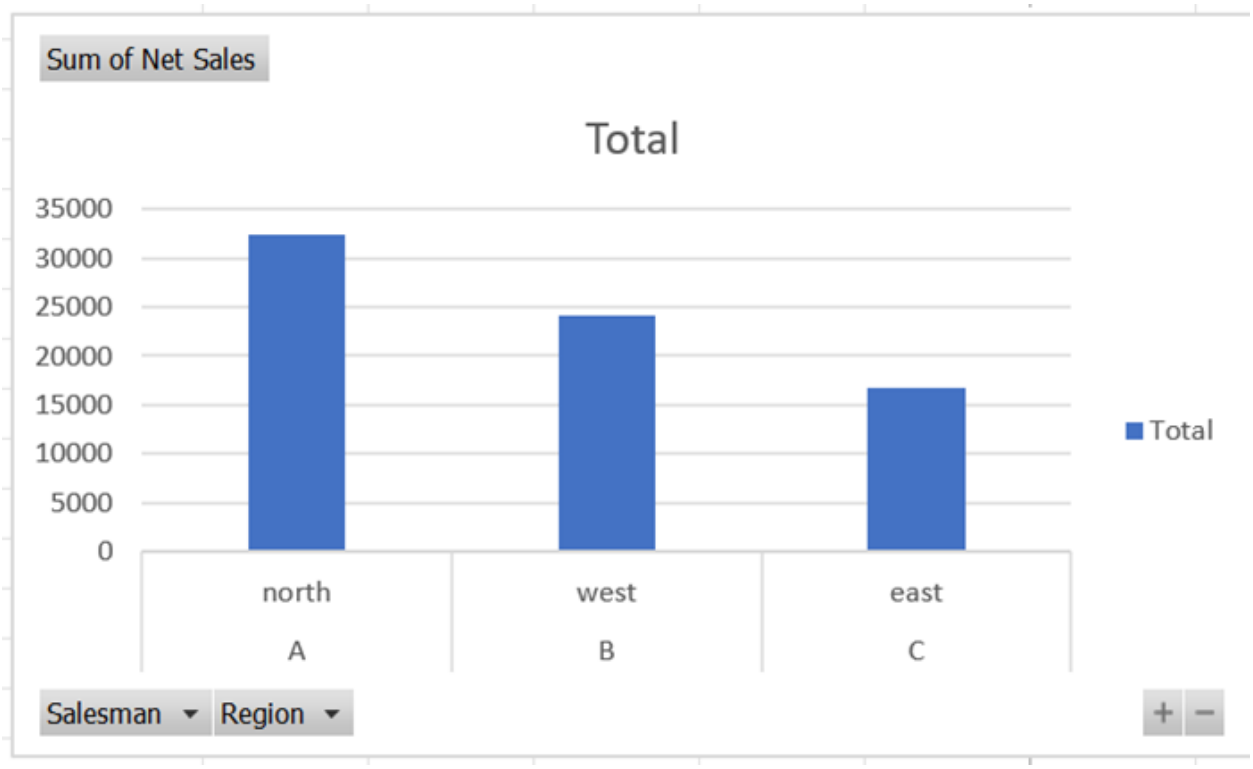
a) Big Data Analysis b) Big Data Charting

Dataset :

	A	B	C	D	E	F	G
1	Month	Salesman	Region	Products	Customers	Net Sales	profit/loss
2	Jan	A	north	car	7	2100	100
3	Jan	A	north	car	8	2400	200
4	Jan	A	north	car	7	2100	200
5	Jan	A	north	car	7	2100	300
6	Jan	A	north	car	6	1800	200
7	Jan	B	west	car	8	2400	100
8	Jan	C	east	car	9	2700	100
9	Jan	C	east	bike	10	2000	100
10	Jan	A	north	bike	6	1200	400
11	Jan	A	north	bike	7	1400	200
12	Jan	A	north	bike	9	1800	100
13	Jan	A	north	bike	5	1000	100
14	Jan	B	west	truck	10	4000	100
15	Jan	B	west	truck	10	4000	200
16	Jan	B	west	truck	10	4000	300
17	Feb	B	west	car	9	2700	400
18	Feb	C	east	car	7	2100	200
19	Feb	C	east	car	6	1800	300
20	Feb	C	east	car	7	2100	300
21	Feb	A	north	car	8	2400	300
22	Feb	A	north	car	9	2700	400
23	Mar	A	north	car	8	2400	400
24	Mar	A	north	car	10	3000	100
25	Mar	A	north	car	5	1500	100
26	Mar	A	north	car	6	1800	100
27	Mar	A	north	car	9	2700	100
28	Mar	B	west	bike	7	1400	200
29	Mar	C	east	car	8	2400	200
30	Mar	B	west	truck	10	4000	300
31	Mar	C	east	truck	9	3600	200
32	Mar	B	west	bike	8	1600	200

Sales by region and product :

Row Labels	Sum of Net Sales
bike	**10400**
east	2000
north	5400
west	3000
car	**43200**
east	11100
north	27000
west	5100
truck	**19600**
east	3600
west	16000
Grand Total	**73200**



Customer count by salesperson :

Row Labels	Sum of Customers
A	117
B	72
C	56
Grand Total	**245**

Q3) Buyer event analytics using Cassandra on suitable product sales data

create keyspace buyer_events with replication = {'class' :
'SimpleStrategy','replication_factor':'1'};

use buyer_events;

BUYERS TABLE :

create table buyers(

... buyer_id int primary key,

... username text,

... email text,

... address text);

Inserting :

insert into buyers(buyer_id,username,email,address)

... values(1,'user1','user1@example.com','hyd');

cqlsh:buyer_events> insert into buyers(buyer_id,username,email,address)

... values(2,'user2','user2@example.com','mum');

cqlsh:buyer_events> insert into buyers(buyer_id,username,email,address)

... values(3,'user3','user3@example.com','pun');

cqlsh:buyer_events> insert into buyers(buyer_id,username,email,address)

... values(4,'user4','user4@example.com','hyd');

cqlsh:buyer_events> insert into buyers(buyer_id,username,email,address)

... values(5,'user5','user5@example.com','pun');

insert into buyers(buyer_id,username,email,address)

... values(6,'user6','user6@example.com','mum');

```
 buyer_id | address | email              | username
----------+---------+--------------------+----------
        5 |     pun | user5@example.com |    user5
        1 |     hyd | user1@example.com |    user1
        2 |     mum | user2@example.com |    user2
        4 |     hyd | user4@example.com |    user4
        6 |     mum | user6@example.com |    user6
        3 |     pun | user3@example.com |    user3

(6 rows)
```

PRODUCTS TABLE :

```
create table products(
            ... product_id int primary key,
            ... name text,
            ... category text,
            ... price int);

cqlsh:buyer_events> insert into products(product_id,name,category,price)
            ... values (1, 'coffee maker', 'appliances', 60);
cqlsh:buyer_events> insert into products (product_id, name, category, price)
            ... values (2, 'phone', 'electronics', 300);
cqlsh:buyer_events> insert into products (product_id, name, category, price)
            ... values (3, 'led tv', 'electronics', 600);
cqlsh:buyer_events> insert into products (product_id, name, category, price)
            ... values (4,'running shoes', 'apparel', 100);
```

```
cqlsh:buyer_events> insert into products (product_id, name, category, price)
               ... values (5,'oven', 'appliances', 500);
cqlsh:buyer_events> select * from products;
```

```
 product_id | category    | name          | price
------------+-------------+---------------+-------
          5 |  appliances |          oven |   400
          1 |  appliances |  coffee maker |    60
          2 | electronics |         phone |   300
          4 |     apparel | running shoes |   100
          3 | electronics |        led tv |   600

(5 rows)
```

PURCHASE HISTORY TABLE :

```
create table purchase_history(
               ... transactionID int primary key,
               ... buyer_id int,
               ... product_id int,
               ... quantity int,
               ... total_amount int,
               ... purchase_date timestamp);
```

INSERTIONS :

```
insert into
purchase_history(transactionID,buyer_id,product_id,quantity,total_amount,purchase_date)
```

... values(1,1,5,2,800,toTimestamp(now()));

cqlsh:buyer_events> insert into purchase_history(transactionID,buyer_id,product_id,quantity,total_amount,purchase_date)

... values(2,1,4,4,400,toTimestamp(now()));

cqlsh:buyer_events> insert into purchase_history(transactionID,buyer_id,product_id,quantity,total_amount,purchase_date)

... values(3,2,4,1,100,toTimestamp(now()));

cqlsh:buyer_events> insert into purchase_history(transactionID,buyer_id,product_id,quantity,total_amount,purchase_date)

... values(4,2,1,3,180,toTimestamp(now()));

cqlsh:buyer_events> insert into purchase_history(transactionID,buyer_id,product_id,quantity,total_amount,purchase_date)

... values(5,3,3,4,2400,toTimestamp(now()));

cqlsh:buyer_events> insert into purchase_history(transactionID,buyer_id,product_id,quantity,total_amount,purchase_date)

... values(6,4,2,2,600,toTimestamp(now()));

transactionid	buyer_id	product_id	purchase_date	quantity	total_amount
5	3	3	2023-11-01 14:52:09+0000	4	2400
1	1	5	2023-11-01 14:50:04+0000	2	800
2	1	4	2023-11-01 14:50:26+0000	4	400
4	2	1	2023-11-01 14:51:31+0000	3	180
6	4	2	2023-11-01 14:52:42+0000	2	600
3	2	4	2023-11-01 14:51:08+0000	1	100

(6 rows)

QUERIES :

1. RETRIEVE BUYER'S PURCHASE HISTORY

```
select * from purchase_history where buyer_id = 1 allow filtering;
```

```
 transactionid | buyer_id | product_id | purchase_date                | quantity | total_amount
---------------+----------+------------+------------------------------+----------+--------------
             1 |        1 |          5 | 2023-11-01 14:50:04+0000 |        2 |          800
             2 |        1 |          4 | 2023-11-01 14:50:26+0000 |        4 |          400

(2 rows)
```

2. FIND TOTAL NUMBER OF PRODUCTS

```
select count(product_id) from products;
```

```
 system.count(product_id)
--------------------------
                        5

(1 rows)
```

3. RETRIEVE TOTAL SPENDING BY A BUYER

```
select sum(total_amount) from purchase_history where buyer_id = 2 allow filtering;
```

```
 system.sum(total_amount)
--------------------------
                      280

(1 rows)
```

4. RETRIEVE PRODUCTS BY CATEGORY

select * from products where category = 'electronics' allow filtering;

```
 product_id | category    | name   | price
------------+-------------+--------+-------
          2 | electronics |  phone |   300
          3 | electronics | led tv |   600

(2 rows)
```

5. FIND THE MAXIMUM PRICE

Select max(price) from products;

```
 system.max(price)
-------------------
               600

(1 rows)
```

6. TO FIND TOTAL NUMBER OF BUYERS

select count(buyer_id) from buyers;

```
 system.count(buyer_id)
------------------------
                      6

(1 rows)
```

7. FIND AVERAGE PRICE OF PRODUCTS OF A PARTICULAR CATEGORY

select avg(price) from products where category='electronics' allow filtering;

```
 system.avg(price)
--------------------
               450

(1 rows)
```

8. SELECT PRODUCTS IN A PARTICULAR PRICE RANGE

select * from products where price>=100 and price<=400 allow filtering;

```
 product_id | category    | name          | price
------------+-------------+---------------+-------
          5 |  appliances |          oven |   400
          2 | electronics |         phone |   300
          4 |     apparel | running shoes |   100

(3 rows)
```

9. TO FIND NUMBER OF BUYERS FROM A SPECIFIC LOCATION

select count(buyer_id) from buyers where address='pun' allow filtering;

```
 system.count(buyer_id)
------------------------
                      2

(1 rows)
```

10. TO FIND ALL THE PURCHASES OF A PARTICULAR PRODUCT

select * from purchase_history where product_id = 4 allow filtering;

```
 transactionid | buyer_id | product_id | purchase_date                | quantity | total_amount
---------------+----------+------------+------------------------------+----------+--------------
             2 |        1 |          4 | 2023-11-01 14:50:26+0000 |        4 |          400
             3 |        2 |          4 | 2023-11-01 14:51:08+0000 |        1 |          100

(2 rows)
```

11. TO FIND TOTAL AMOUNT OBTAINED THROUGH SALES OF A PARTICULAR PRODUCT

select sum(total_amount) from purchase_history where product_id = 4 allow filtering;

```
 system.sum(total_amount)
--------------------------
                      500

(1 rows)
```

Q4) Perform Social media analysis using cassandra

```
create keyspace social_media
   ... with replication = {'class':'SimpleStrategy','replication_factor':1};
use social_media;
```

USERS TABLE :

```
create table social_media.users(
        ... user_id int primary key,
        ... username text,
        ... email text,
        ... registration_date timestamp
        ... );
```

INSERTING INTO USERS :

```
insert into social_media.users(user_id,username,email,registration_date)
        ... values(1,'user1','user1@example.com',toTimestamp(now()));
insert into social_media.users(user_id,username,email,registration_date)
        ... values(2,'user2','user2@example.com',toTimestamp(now()));
insert into social_media.users(user_id,username,email,registration_date)
        ... values(3,'user3','user3@example.com',toTimestamp(now()));
insert into social_media.users(user_id,username,email,registration_date)
        ... values(4,'user4','user4@example.com',toTimestamp(now()));
insert into social_media.users(user_id,username,email,registration_date)
        ... values(5,'user5','user5@example.com',toTimestamp(now()));
```

USERS TABLE :

select * from users;

user_id	email	registration_date	username
5	user5@example.com	2023-10-31 16:27:59+0000	user5
1	user1@example.com	2023-10-31 16:24:52+0000	user1
2	user2@example.com	2023-10-31 16:26:41+0000	user2
4	user4@example.com	2023-10-31 16:27:46+0000	user4
3	user3@example.com	2023-10-31 16:27:29+0000	user3

(5 rows)

Q5) Use R-project to carry out statistical analysis of big data

Perform analytics on any standard data set

```
dat <- read.csv(file =

"C:\\Users\\Dakshith\\Downloads\\Compressed\\Datasets\\inflammation-01.csv", header =

FALSE)

head(dat)
```

output:

```
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21
1  0  0  1  3  1  2 4  7  8  3  3  3  10  5  7  4  7  7 12  18  6
2  0  1  2  1  2  1  3  2  2  6  10  11  5  9  4  4  7  16  8  6  18
3  0  1  1  3  3  2  6  2  5  9  5  7  4  5  4  15  5  11  9  10  19
4  0  0  2  0  4  2  2  1  6  7  10  7  9  13  8  8  15  10  10  7  17
5  0  1  1  3  3  1  3  5  2  4  4  7  6  5  3  10  8  10  6  17  9
6  0  0  1  2  2  4  2  1  6  4  7  6  6  9  9  15  4  16  18  12  12
  V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40
1 13  11  11  7  7  4  6  8  8  4  4  5  7  3  4  2  3  0  0
2  4  12  5  12  7  11  5  11  3  3  5  4  4  5  5  1  1  0  1
3  14  12  17  7  12  11  7  4  2  10  5  4  2  2  3  2  2  1  1
4  4  4  7  6  15  6  4  9  11  3  5  6  3  3  4  2  3  2  1
5  14  9  7  13  9  12  6  7  7  9  6  3  2  2  4  2  0  1  1
6  5  18  9  5  3  10  3  12  7  8  4  7  3  5  4  4  3  2  1
```

```
patient_1 <- dat[1, ]

# max inflammation for patient 1

max(patient_1)
```

```
[1] 18
```

```
max(dat[2, ])
```

```
18
```

```
min(dat[, 7])
```

```
1
```

```
mean(dat[, 7])
```

```
3.8
```

```
sd(dat[,7])
```

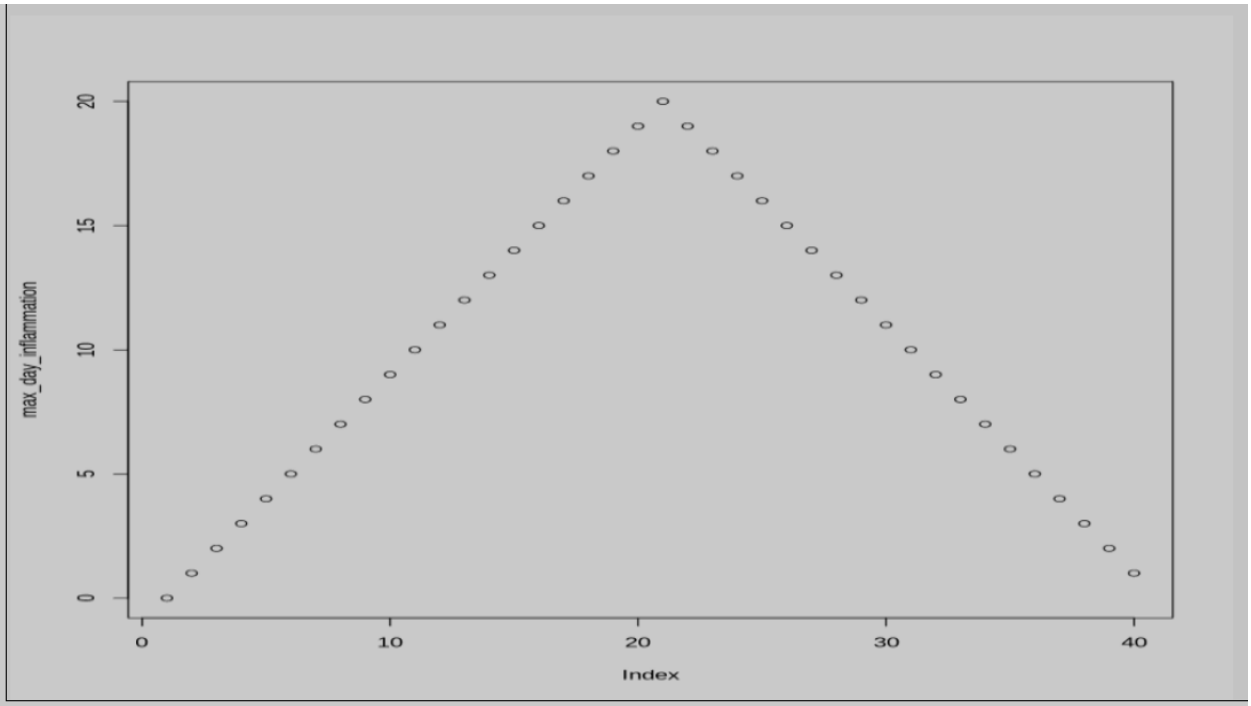
```
[1] 1.725187
```

```
summary(dat[,1:4])
```

```
       V1              V2              V3              V4
 Min.   :0.000   Min.   :0.00   Min.   :0.000   Min.   :0.00
 1st Qu.:0.000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:1.00
 Median :0.000   Median :0.00   Median :1.000   Median :2.00
 Mean   :0.000   Mean   :0.45   Mean   :1.117   Mean   :1.75
 3rd Qu.:0.000   3rd Qu.:1.00   3rd Qu.:2.000   3rd Qu.:3.00
 Max.   :0.000   Max.   :1.00   Max.   :2.000   Max.   :3.00
```

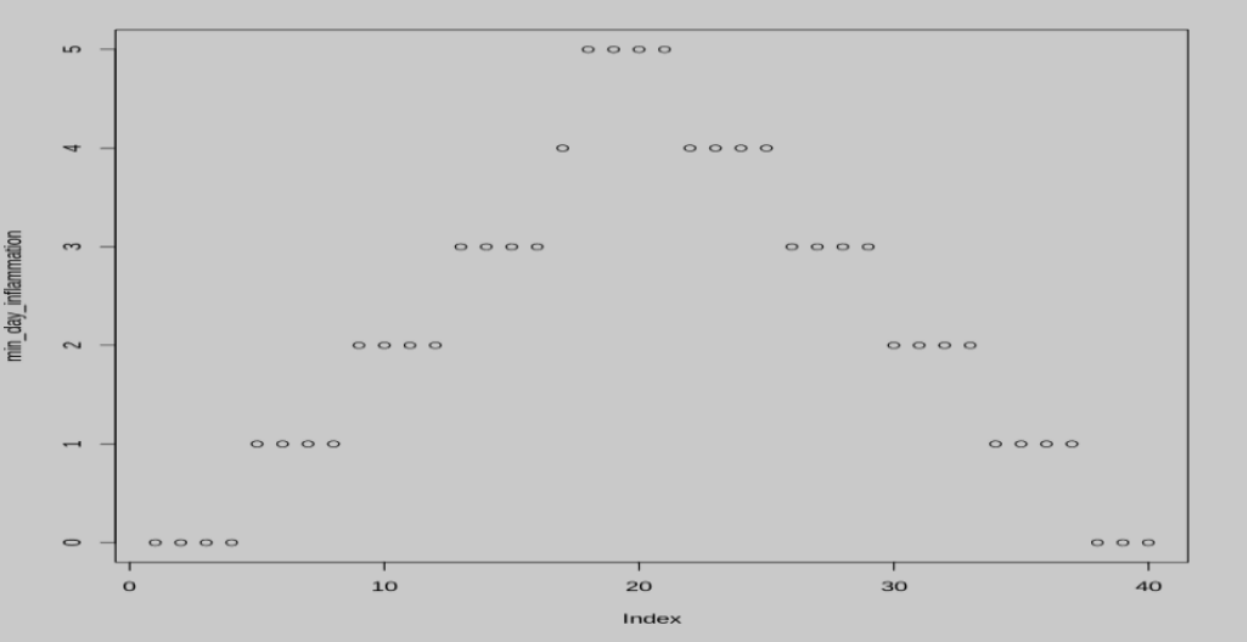
```
avg_day_inflammation<-apply(dat,2,mean)
```

```
plot(avg_day_inflammation)
```



min_day_inflammation<-apply(dat,2,min)

plot(min_day_inflammation)



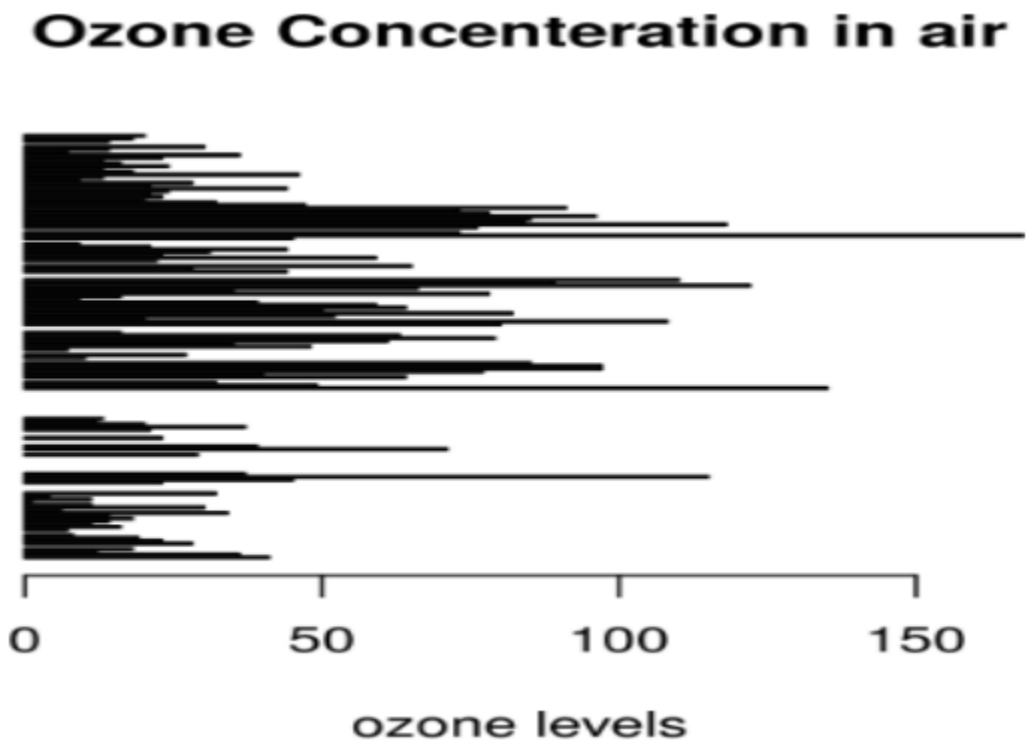
Q6) Use R-Project for data visualization of social media data

Bar Plot:

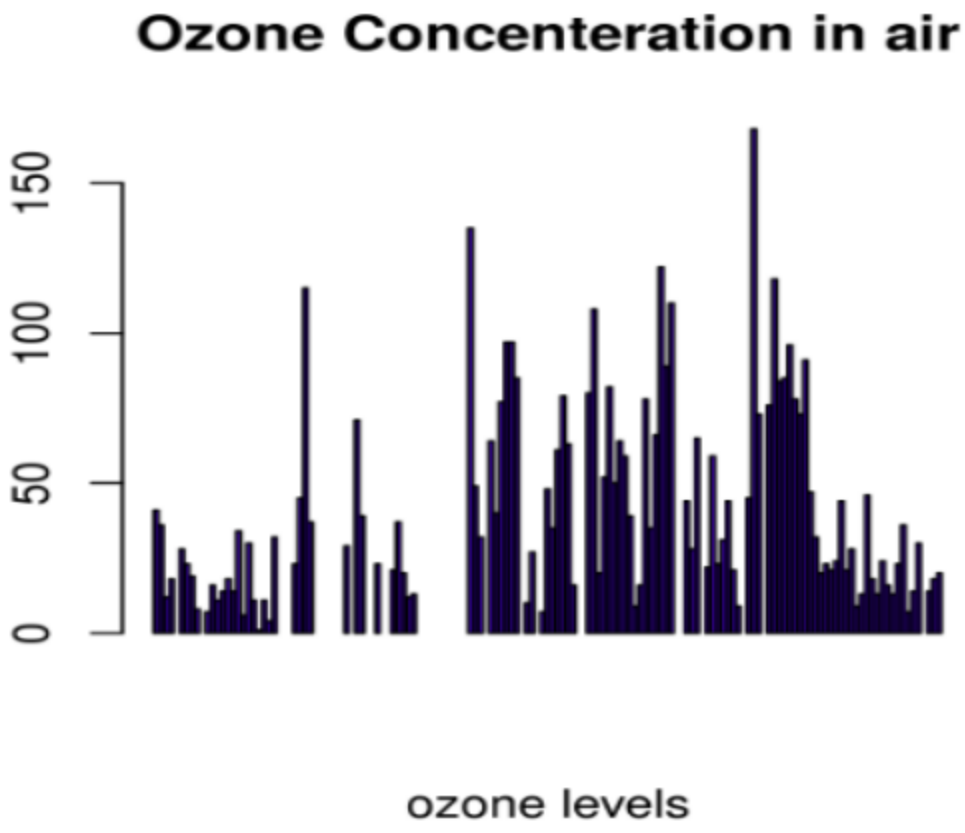
data(airquality)

barplot(airquality$Ozone, main = 'Ozone Concenteration in air',

xlab = 'ozone levels', horiz = TRUE)

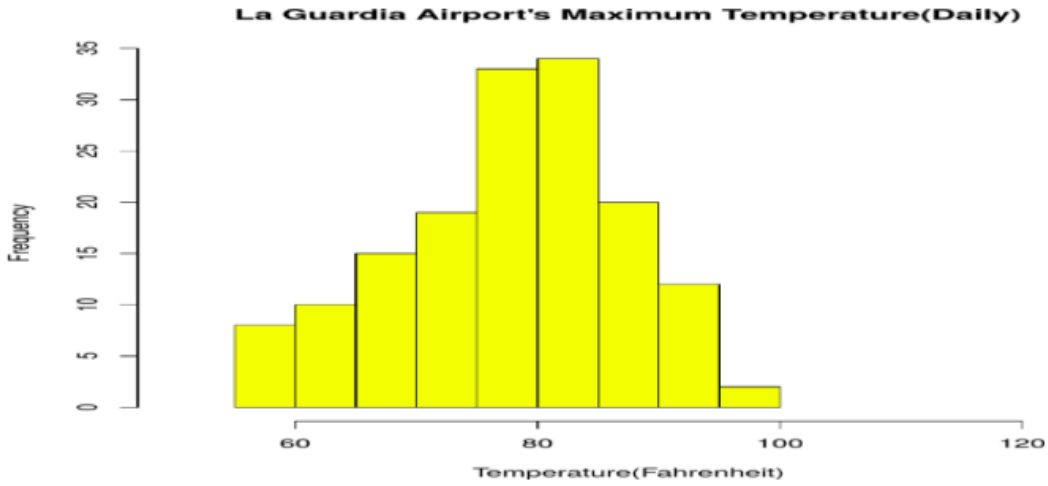


barplot(airquality$Ozone, main = 'Ozone Concenteration in air', xlab = 'ozone levels', col ='blue', horiz = FALSE)



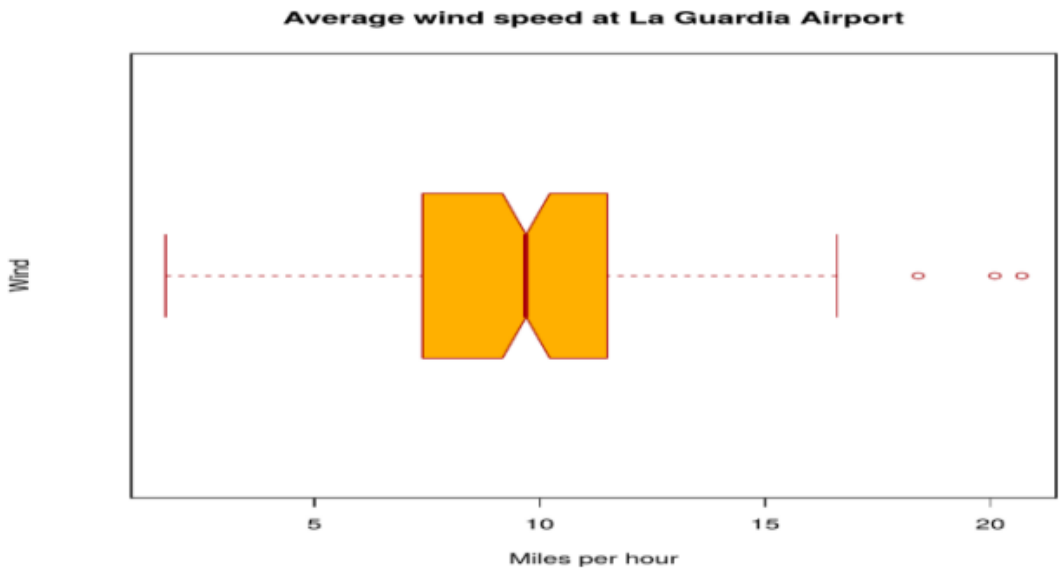
Histogram:

hist(airquality$Temp, main ="La Guardia Airport's Maximum Temperature(Daily)", xlab ="Temperature(Fahrenheit)", xlim = c(50, 125), col ="yellow", freq = TRUE)

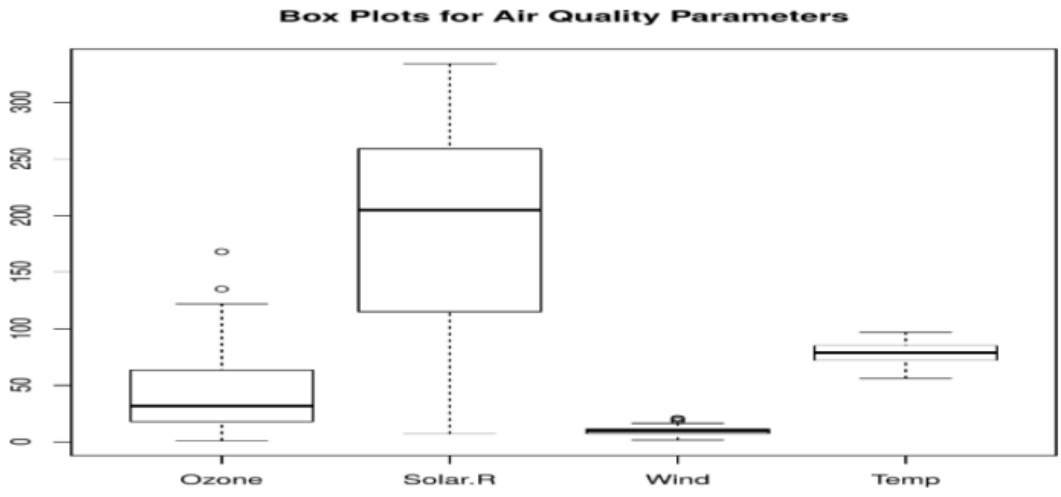


Box Plot:

boxplot(airquality$Wind, main = "Average wind speed at La Guardia Airport", xlab = "Miles per hour", ylab = "Wind", col = "orange", border = "brown", horizontal = TRUE, notch = TRUE)

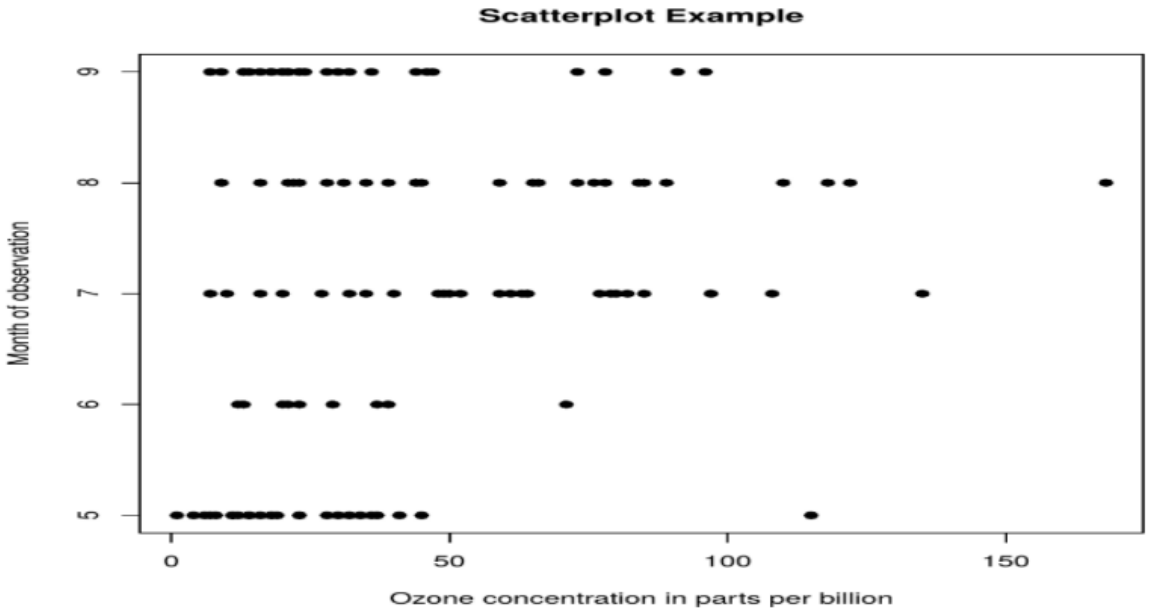


boxplot(airquality[, 0:4], main ='Box Plots for Air Quality Parameters')



Scatter plot:

plot(airquality$Ozone, airquality$Month, main ="Scatterplot Example", xlab ="Ozone Concentration in parts per billion", ylab =" Month of observation ", pch = 19)


Scatterplot Example
Month of observation
9
8
7
6
5
0
50
100
150
Ozone concentration in parts per billion