# 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]));

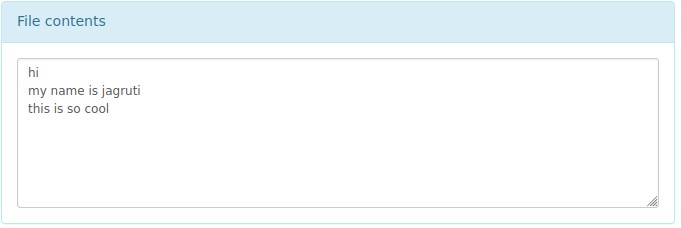
FileOutputFormat.setOutputPath(conf,new Path(args[1]));

JobClient.runJob(conf);

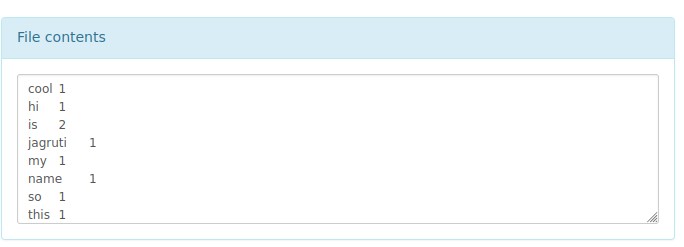
}

}

# Input file

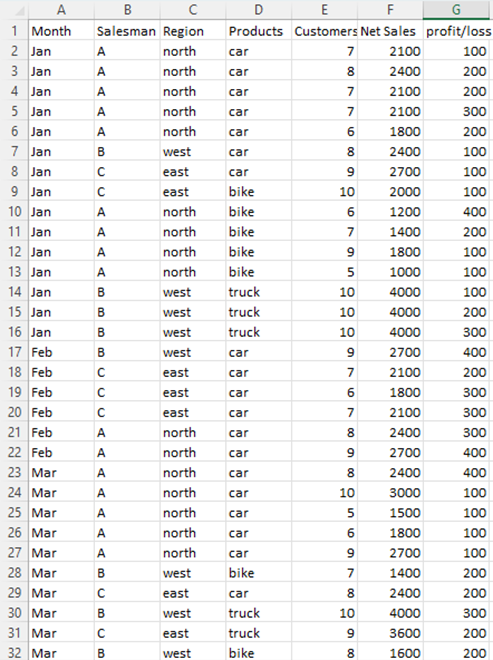


# Output file

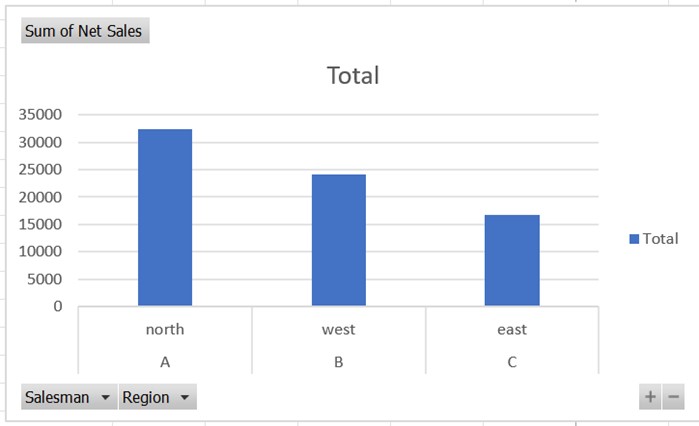
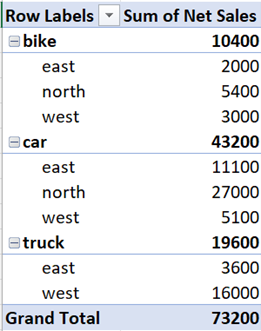


# Q2)Using Power Pivot (Excel) Perform the following on any dataset

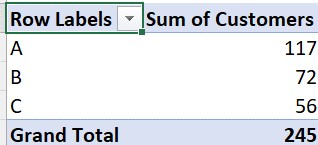
**a)** **Big Data Analysis b) Big Data Charting Dataset :**



**Sales by region and product :**



**Customer count by salesperson :**



# 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)

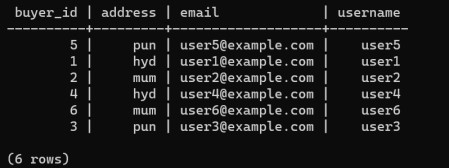
... values(2,'user2','user2@example.com','mum'); cqlsh:buyer\_events> insert into buyers(buyer\_id,username,email,address)

... 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');



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)

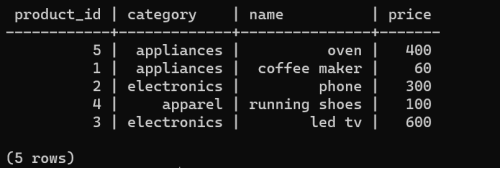
... values (2, 'phone', 'electronics', 300); cqlsh:buyer\_events> insert into products (product\_id, name, category, price)

... 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;



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()));

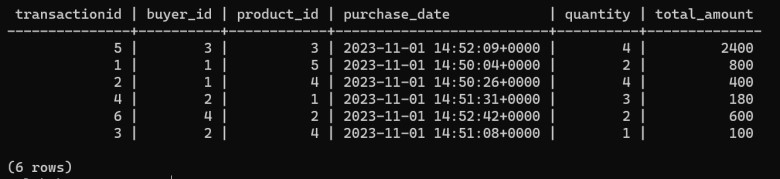
cqlsh:buyer\_events> insert into purchase\_history(transactionID,buyer\_id,product\_id,quantity,total\_amount,purchase\_date)

... 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()));

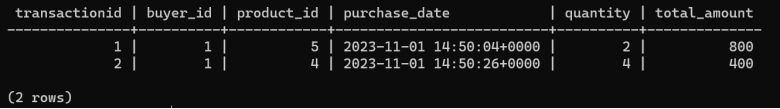
cqlsh:buyer\_events> insert into purchase\_history(transactionID,buyer\_id,product\_id,quantity,total\_amount,purchase\_date) ... values(6,4,2,2,600,toTimestamp(now()));



QUERIES :

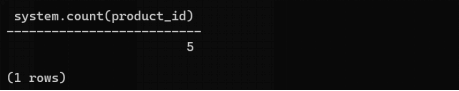
1. RETRIEVE BUYER’S PURCHASE HISTORY

select \* from purchase\_history where buyer\_id = 1 allow filtering;



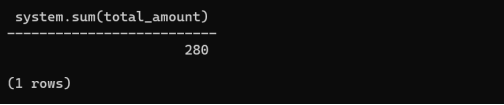
1. FIND TOTAL NUMBER OF PRODUCTS

select count(product\_id) from products;



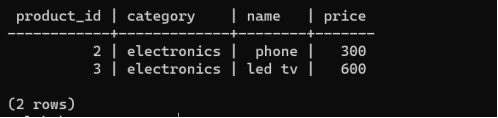
1. RETRIEVE TOTAL SPENDING BY A BUYER

select sum(total\_amount) from purchase\_history where buyer\_id = 2 allow filtering;



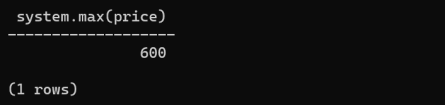
1. RETRIEVE PRODUCTS BY CATEGORY

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



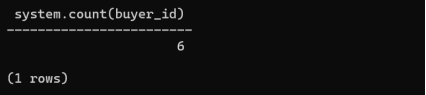
1. FIND THE MAXIMUM PRICE

Select max(price) from products;



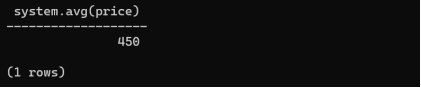
1. TO FIND TOTAL NUMBER OF BUYERS

select count(buyer\_id) from buyers;



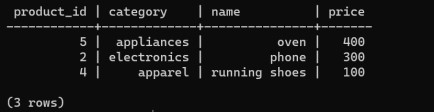
1. FIND AVERAGE PRICE OF PRODUCTS OF A PARTICULAR CATEGORY

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



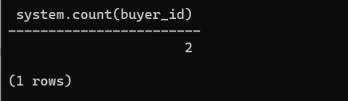
1. SELECT PRODUCTS IN A PARTICULAR PRICE RANGE

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



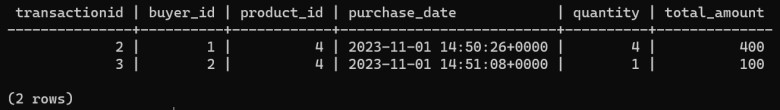
1. TO FIND NUMBER OF BUYERS FROM A SPECIFIC LOCATION

select count(buyer\_id) from buyers where address='pun' allow filtering;



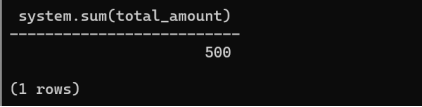
1. TO FIND ALL THE PURCHASES OF A PARTICULAR PRODUCT

select \* from purchase\_history where product\_id = 4 allow filtering;



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

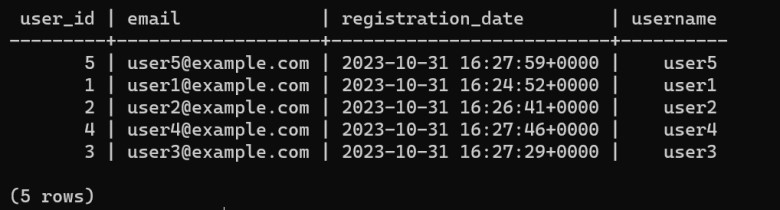
select sum(total\_amount) from purchase\_history where product\_id = 4 allow filtering;



|  |  |
| --- | --- |
| 15    **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()));  Section-A BDA Lab | JNTUH UCESTH |

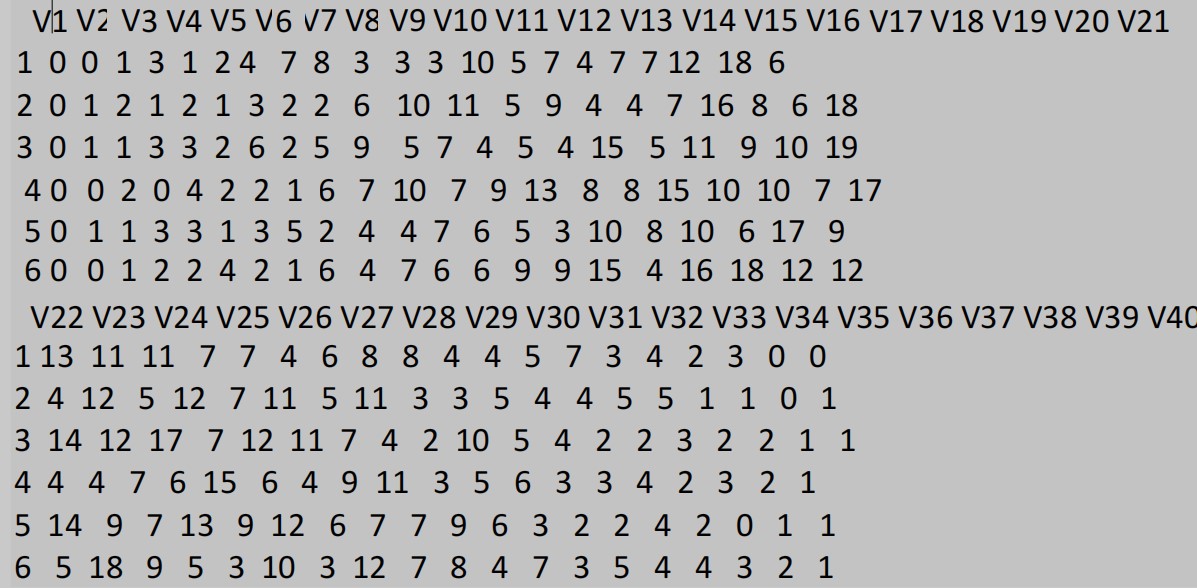
USERS TABLE :

select \* from users;



# 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:



patient\_1 <- dat[1, ]

# max inflammation for patient 1 max(patient\_1)



max(dat[2, ])



min(dat[, 7])



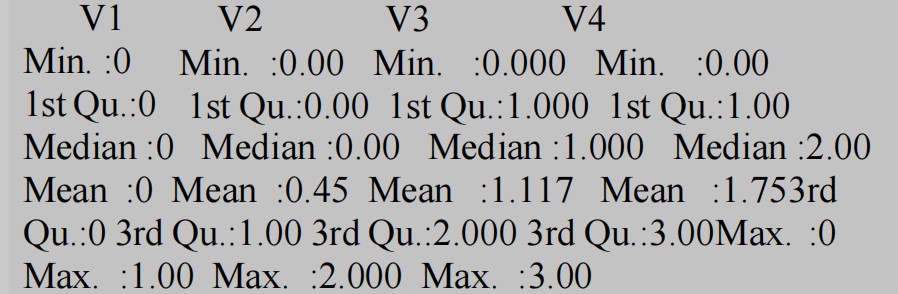
mean(dat[, 7])



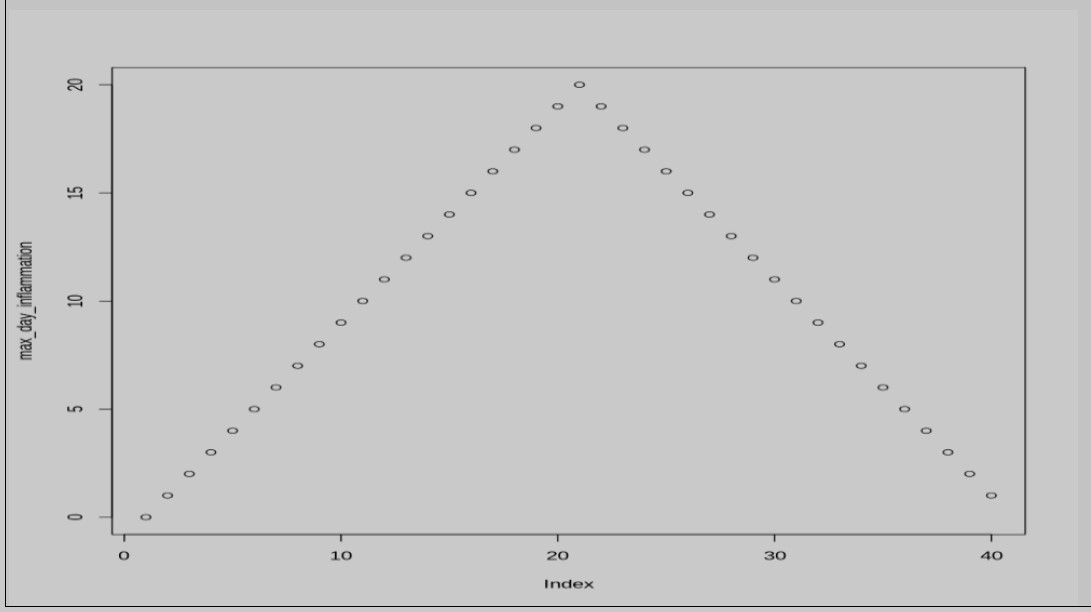
sd(dat[,7])



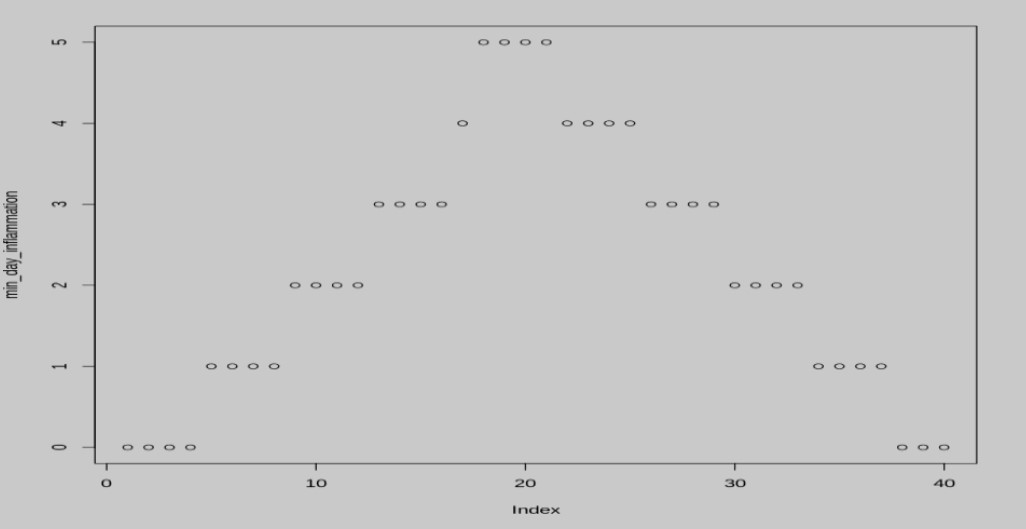
summary(dat[,1:4])



avg\_day\_inflammation<-apply(dat,2,mean) plot(avg\_day\_inflammation)



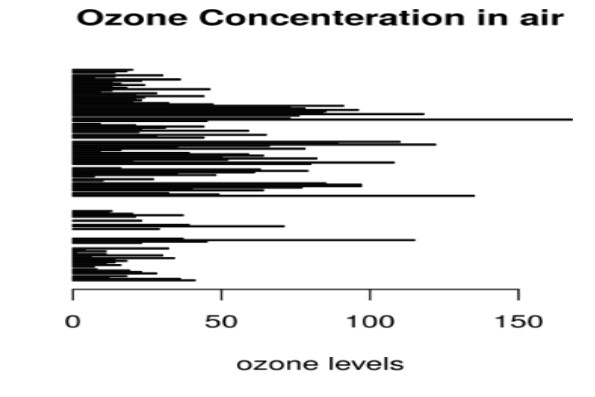
min\_day\_inflammation<-apply(dat,2,min) plot(min\_day\_inflammation)



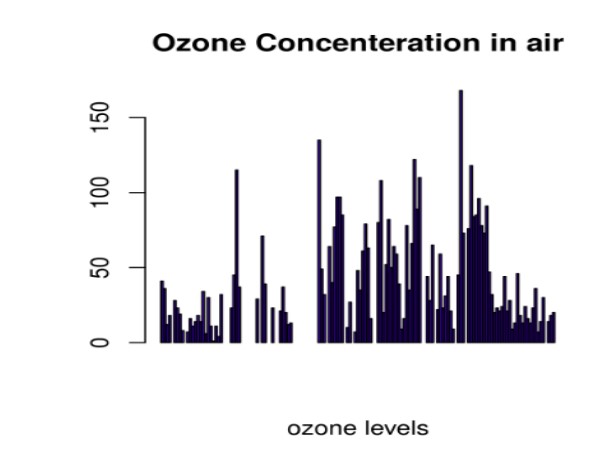
**Q6) Use R-Project for data visualization of social media data**

Bar Plot:

data(airquality) barplot(airquality$Ozone, main = 'Ozone Concenteration in air', xlab = 'ozone levels', horiz = TRUE)



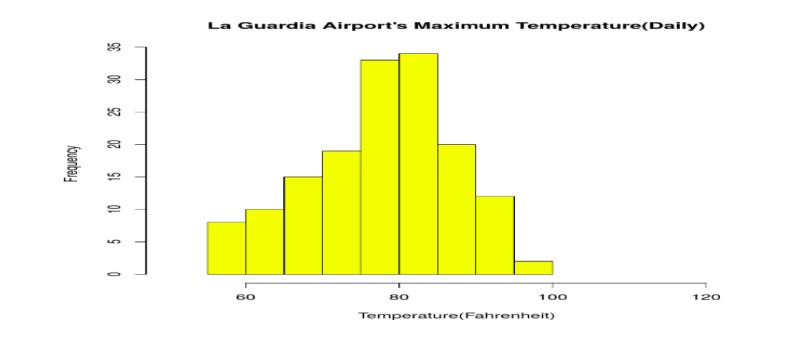
barplot(airquality$Ozone, main = 'Ozone Concenteration in air', xlab = 'ozone levels', col ='blue', horiz = FALSE)



Histogram:

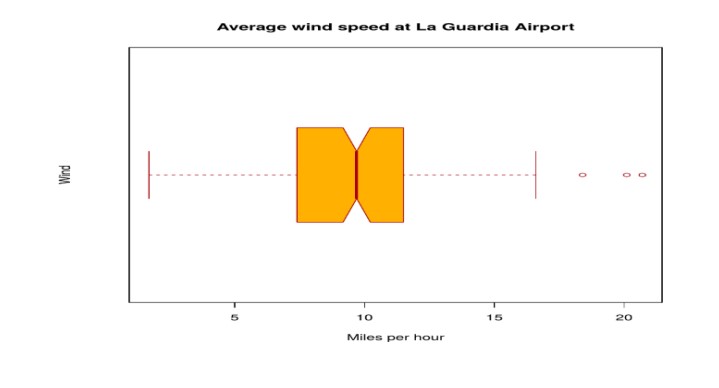
hist(airquality$Temp, main ="La Guardia Airport's Maximum Temperature(Daily)", xlab

="Temperature(Fahrenheit)", xlim = c(50, 125), col ="yellow", freq = TRUE)

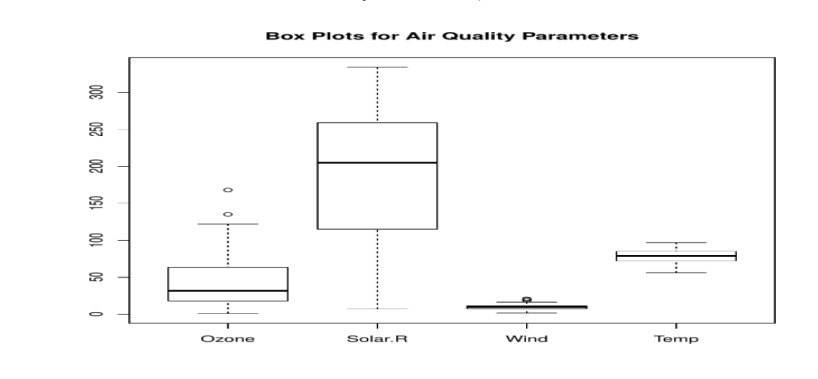


Box Plot:

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)

