



University of Mumbai

PRACTICAL JOURNAL

PSDS2P3: Algorithm For Data Science

**(MSc. Computer Science with specialization in Data Science
2021-2023)**

SEM-II

Submitted by

Sumon Singh

Roll No. 16

Seat No. 112011

Problems :

- 1. Perform the operations in Hadoop HDFS**
- 2. Create a wordcount application in MapReduce**
- 3. Implement KMeans algorithm**
- 4. Implement Hierarchical clustering**
- 5. Implement KNN classifier**
- 6. Implement KNN regressor**
- 7. Time series**
- 8. Predicting the authors of the disputed federalist papers**
- 9. Implement Sentiment analysis**
- 10. Perform regression analysis with any of the datasets**

Name : Sumon Singh

**Course : M.sc Computer Science With Specialization In
Data Science**

Paper : PSDS203 (Algorithms For Data Science)

Assignment : 1

Tasks :

Perform the following operations in Hadoop HDFS,take screenshots and upload pdf.

You should run the commands from your user name.

User name should be displayed at the prompt.

Write short description in your own words about the functionality you are performing.

1. Start Hadoop Namenode and datanode

Description :

User name is hadoop .

Namenode is the master node which manages all the data nodes and it executes the requirements of client such as data store, data retrieve from data nodes.

Data node are slave nodes where physically the data is being stored.

```
sumon@Lenovo:/home/hadoop$ su hadoop
Password:
hadoop@Lenovo:~$ echo $USER
hadoop
hadoop@Lenovo:~$ start-dfs.sh
Starting namenodes on [192.168.0.107]
Starting datanodes
Starting secondary namenodes [Lenovo]
hadoop@Lenovo:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

2. Print your Hadoop version

Description :

`hadoop version` command shows what version of hadoop is currently installed in the system which is 3.3.0.

```
hadoop@Lenovo:~$ hadoop version
Hadoop 3.3.0
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r aa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled by brahma on 2020-07-06T18:44Z
Compiled with protoc 3.7.1
From source with checksum 5dc29b802d6ccd77b262ef9d04d19c4
This command was run using /home/hadoop/hadoop/share/hadoop/common/hadoop-common-3.3.0.jar
```

3. Print the nodes which are running

Description :

`jps` command shows which nodes are currently running for the hadoop file system.

```
hadoop@Lenovo:~$ jps
21858 SecondaryNameNode
21444 NameNode
22087 ResourceManager
22776 Jps
22266 NodeManager
21631 DataNode
```

4. Create a directory Hadoop-assign-1 in HDFS

Description :

`hadoop fs mkdir /Hadoop-assign-1` command creates a directory named Hadoop-assign-1 in the hadoop file system at home (/) location.

```
hadoop@Lenovo:~$ hadoop fs -mkdir /Hadoop-assign-1
hadoop@Lenovo:~$ hadoop fs -ls /
Found 1 items
drwxr-xr-x  - hadoop supergroup          0 2022-03-20 13:27 /Hadoop-assign-1
```

5. Create an empty file my-hadoop-assign.txt

Description :

`hadoop fs touchz /Hadoop-assign-1/my-hadoop-assign.txt` command creates an empty text file at /Hadoop-assign-1 location.

```
hadoop@Lenovo:~$ hadoop fs -touchz /Hadoop-assign-1/my-hadoop-assign.txt
hadoop@Lenovo:~$ hadoop fs -ls -R /
drwxr-xr-x  - hadoop supergroup          0 2022-03-20 13:28 /Hadoop-assign-1
-rw-r--r--  1 hadoop supergroup          0 2022-03-20 13:28 /Hadoop-assign-1/my-hadoop-assign.txt
```

6. Create a file called “hare_story.txt” in C:\ with the content copied from the story Hare And Tortoise Story Bedtimeshortstories

Description :

In linux system, create a hare_story.txt text file at /home directory and copied the asked story.

```
hadoop@Lenovo:~$ touch hare_story.txt
```

7. Append the content from the local file “hare_story.txt” to “my-hadoop-assign.txt” in HDFS.

Description :

The following command append the content of the hare_story.txt file which is in local system to /Hadoop-assign-1/my-hadoop-assign.txt of hadoop file system.

```
hadoop@Lenovo:~$ hadoop fs -appendToFile hare_story.txt /Hadoop-assign-1/my-hadoop-assign.txt
```

8. Copy any local file to HDFS directly.

Description :

In the local system a file named dummy.txt is created and then it is being copied to the hadoop file system.

```
hadoop@Lenovo:~$ hadoop fs -put dummy.txt /
hadoop@Lenovo:~$ hadoop fs -ls -R /
drwxr-xr-x  - hadoop supergroup          0 2022-03-20 13:28 /Hadoop-assign-1
-rw-r--r--  1 hadoop supergroup      1298 2022-03-20 13:29 /Hadoop-assign-1/my-hadoop-assign.txt
-rw-r--r--  1 hadoop supergroup       61 2022-03-20 13:30 /dummy.txt
```

9. View the content of HDFS with subfolders and files.

Description :

The following command show all the directories and sub-directories of the hadoop file system.

```
hadoop@Lenovo:~$ hadoop fs -ls -R /
drwxr-xr-x  - hadoop supergroup          0 2022-03-20 13:28 /Hadoop-assign-1
-rw-r--r--  1 hadoop supergroup      1298 2022-03-20 13:29 /Hadoop-assign-1/my-hadoop-assign.txt
-rw-r--r--  1 hadoop supergroup       61 2022-03-20 13:30 /dummy.txt
```

10. Rename the “my-hadoop-assign.txt” to “my-homework.txt”

Description :

The following command renames a file name of hadoop file system. For verifying the change list of all the directories and sub-directories are shown.

```
hadoop@Lenovo:~$ hadoop fs -mv /Hadoop-assign-1/my-hadoop-assign.txt /Hadoop-assign-1/my-homework.txt
hadoop@Lenovo:~$ hadoop fs -ls -R /
drwxr-xr-x  - hadoop supergroup          0 2022-03-20 13:32 /Hadoop-assign-1
-rw-r--r--  1 hadoop supergroup      1298 2022-03-20 13:29 /Hadoop-assign-1/my-homework.txt
-rw-r--r--  1 hadoop supergroup       61 2022-03-20 13:30 /dummy.txt
```

11. View the size of the files in HDFS in terms of KB .

Description :

The following command shows each of the file sizes of hadoop file system.

```
hadoop@Lenovo:/home/sumon$ hadoop fs -du -h /
1.3 K  1.3 K  /Hadoop-assign-1
61      61      /dummy.txt
```

12. Show the web UI with the files in storage.

Description :

This is the web UI of the hadoop file storage which is being accessed by http://ip-address/9870

Browsing HDFS

Sun Mar 20 13:34

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hadoop	supergroup	0 B	Mar 20 13:32	0	0 B	Hadoop-assign-1
-rw-r--r--	hadoop	supergroup	61 B	Mar 20 13:30	1	128 MB	dummy.txt

Showing 1 to 2 of 2 entries

Previous 1 Next

Hadoop, 2020.

Name : Sumon Singh

**Course : M.sc Computer Science With Specialization In
Data Science**

Paper : PSDS203 (Algorithms For Data Science)

Assignment : 2

Task :

Create a wordcount application in MapReduce using Apache pig

Pre-requisite :

- 1. System : Linux (Ubuntu 20.04 LTS)**
- 2. Install and setup hadoop and apache pig**
- 3.**

A. Start hadoop dfs and yarn :

```
start-dfs.sh  
start-yarn.sh
```

B. Start apache pig :

```
pig -x local
```

Steps For WordCount application :

Step 1 : Read each line of a text file from hdfs and store in a variable

Command :

```
lines = load 'hdfs://localhost:9000/hare.txt' using TextLoader as (line:chararray);  
  
DUMP lines;
```

Output :

```

hadoop@Lenovo: ~
hadoop@Lenovo: ~
hadoop@Lenovo: /home/sumon

HadoopVersion PigVersion UserId StartedAt FinishedAt Features
3.3.0 0.17.0 hadoop 2022-04-24 19:03:54 2022-04-24 19:03:55 UNKNOWN

Success!

Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceT
ime AvgReduceTime MedianReduceTime Alias Feature Outputs
job_local1442836923_0007 1 0 n/a n/a n/a n/a 0 0 0 0 0 li
nes MAP_ONLY file:/tmp/temp-1809641881/tmp171827152,

Input(s):
Successfully read 11 records (6860 bytes) from: "hdfs://localhost:9000/hare.txt"

Output(s):
Successfully stored 11 records in: "file:/tmp/temp-1809641881/tmp171827152"

Counters:
Total records written : 11
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1442836923_0007

2022-04-24 19:03:55,241 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system

```

```

hadoop@Lenovo: ~
hadoop@Lenovo: ~
hadoop@Lenovo: /home/sumon

2022-04-24 19:03:55,241 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system
already initialized!
2022-04-24 19:03:55,243 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system
already initialized!
2022-04-24 19:03:55,245 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system
already initialized!
2022-04-24 19:03:55,247 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLauncher - Success!
2022-04-24 19:03:55,248 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is d
eprecated. Instead, use dfs.bytes-per-checksum
2022-04-24 19:03:55,249 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been
initialized
2022-04-24 19:03:55,251 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to
process : 1
2022-04-24 19:03:55,251 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input p
aths to process : 1
(The Hare & the Tortoise A Hare was making fun of the Tortoise one day for being so slow. "Do)
(you ever get anywhere?" he asked with a mocking laugh. "Yes," replied the Tortoise, "and I get)
(there sooner than you think. I'll run you a race and prove it." The Hare was much amused at the)
(idea of running a race with the Tortoise, but for the fun of the thing he agreed. So the Fox, who)
(had consented to act as judge, marked the distance and started the runners off. The Hare was)
(so far out of sight, and to make the Tortoise feel very deeply how ridiculous it was for him to)
(try a race with a Hare, he lay down beside the course to take a nap until the Tortoise should)
(catch up. The Tortoise meanwhile kept going slowly but steadily, and, after a time, passed the)
(place where the Hare was sleeping. But the Hare slept on very peacefully; and when at last he)
(did wake up, the Tortoise was near the goal. The Hare now ran his swiftest, but he could not)
(overtake the Tortoise in time.)
grunt>

```

Comment : Load a text file named “hare.txt” from hdfs and storing each of the lines of the file in lines variable.

Step 2 : Separate each words of each line into tokens (tuples)

Command :

```
words = foreach lines generate flatten(TOKENIZE(line)) as word;
```

```
DUMP words;
```

Output :

```
hadoop@Lenovo: ~          hadoop@Lenovo: /home/sumon
2022-04-24 19:06:28,442 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:
HadoopVersion  PigVersion      UserId  StartedAt      FinishedAt      Features
3.3.0    0.17.0  hadoop  2022-04-24 19:06:28  2022-04-24 19:06:28  UNKNOWN

Success!

Job Stats (time in seconds):
JobId   Maps   Reduces  MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime     MaxReduceTime   MinReduceT
ime    AvgReduceTime  MedianReducetime   Alias  Feature Outputs
job_local1285035797_0008       1        0      n/a      n/a      n/a      n/a      0        0        0        0      li
nes,words      MAP_ONLY           file:/tmp/temp-1809641881/tmp928673734,

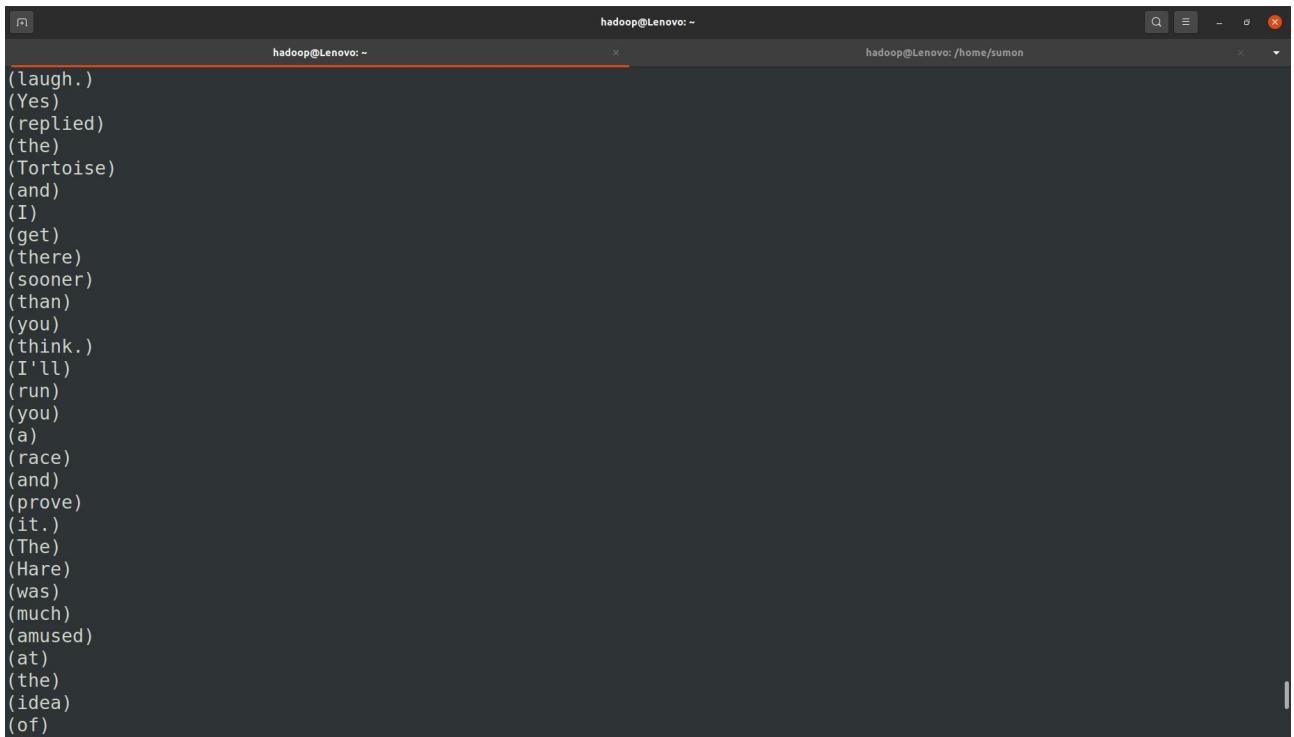
Input(s):
Successfully read 11 records (7840 bytes) from: "hdfs://localhost:9000/hare.txt"

Output(s):
Successfully stored 193 records in: "file:/tmp/temp-1809641881/tmp928673734"

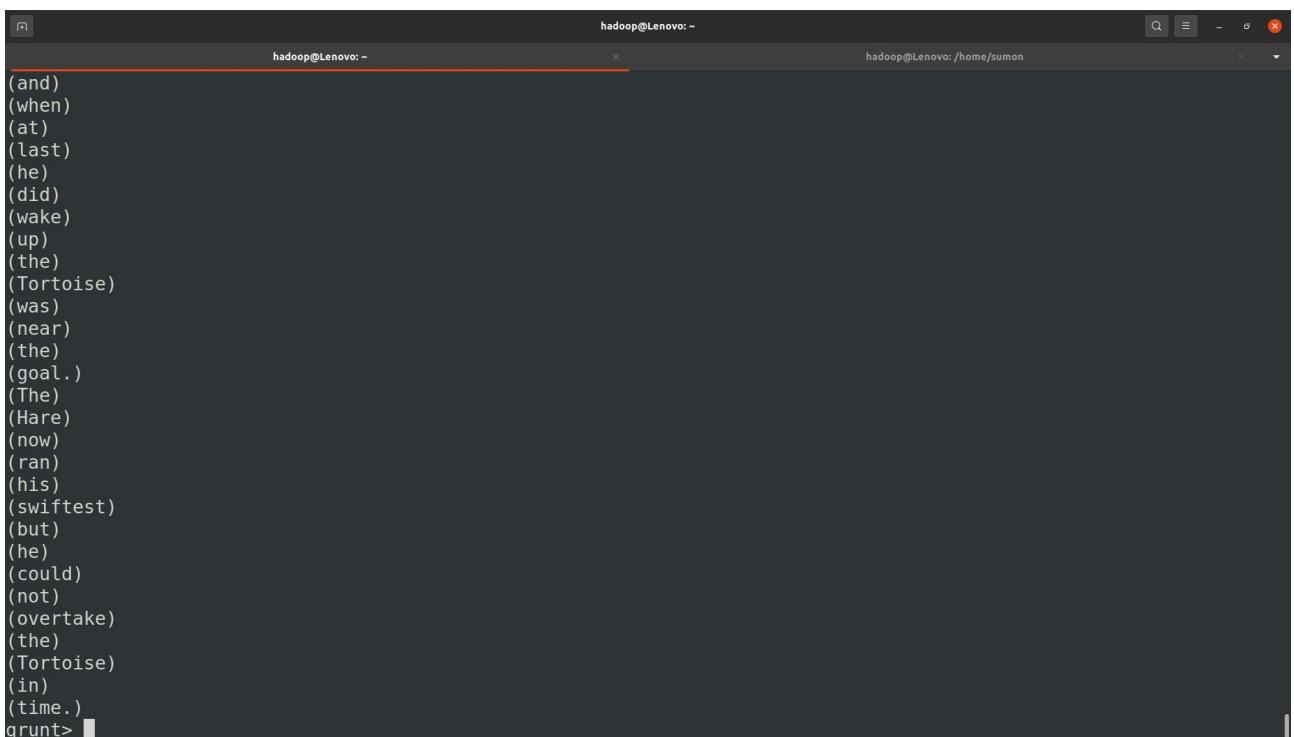
Counters:
Total records written : 193
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1285035797_0008
```

```
hadoop@Lenovo: ~          hadoop@Lenovo: /home/sumon
aths to process : 1
(The)
(Hare)
(&)
(the)
(Tortoise)
(A)
(Hare)
(was)
(making)
(fun)
(of)
(the)
(Tortoise)
(one)
(day)
(for)
(being)
(so)
(slow.)
(Do)
(you)
(ever)
(get)
(anywhere?)
(he)
(asked)
(with)
(a)
(mocking)
```



```
(laugh.)
(Yes)
(replied)
(the)
(Tortoise)
(and)
(I)
(get)
(there)
(sooner)
(than)
(you)
(think.)
(I'll)
(run)
(you)
(a)
(race)
(and)
(prove)
(it.)
(The)
(Hare)
(was)
(much)
(amused)
(at)
(the)
(idea)
(of)
```



```
(and)
(when)
(at)
(last)
(he)
(did)
(wake)
(up)
(the)
(Tortoise)
(was)
(near)
(the)
(goal.)
(The)
(Hare)
(now)
(ran)
(his)
(swiftest)
(but)
(he)
(could)
(not)
(overtake)
(the)
(Tortoise)
(in)
(time.)
grunt> █
```

Comment : In words variable store each word of each line individually as a tuple.

Command :

Step 3 : Make the similar words in the same group

groups = group words by word;

DUMP groups;

Output :

```
hadoop@Lenovo: ~          hadoop@Lenovo: ~          hadoop@Lenovo: /home/sumon
HadoopVersion  PigVersion     UserId  StartedAt      FinishedAt    Features
3.3.0   0.17.0  hadoop  2022-04-24 19:08:11  2022-04-24 19:08:11  GROUP_BY

Success!

Job Stats (time in seconds):
JobId  Maps   Reduces  MaxMapTime   MinMapTime   AvgMapTime   MedianMapTime  MaxReduceTime  MinReduceT
ime   AvgReduceTime  MedianReduceTime  Alias  Feature Outputs
job_local1985066520_0009  1       1       n/a        n/a        n/a        n/a        n/a        n/a        n/a        gr
oups,lines,words          GROUP_BY           file:/tmp/temp-1809641881/tmp-88083735,

Input(s):
Successfully read 11 records (17640 bytes) from: "hdfs://localhost:9000/hare.txt"

Output(s):
Successfully stored 116 records in: "file:/tmp/temp-1809641881/tmp-88083735"

Counters:
Total records written : 116
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1985066520_0009

2022-04-24 19:08:11,365 [main] WARN  org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system | already initialized!
```

```
hadoop@Lenovo: ~          hadoop@Lenovo: ~          hadoop@Lenovo: /home/sumon
(amused,{{(amused)}})
(beside,{{(beside)}})
(course,{{(course)}})
(deeply,{{(deeply)}})
(laugh,{{(laugh.)}})
(making,{{(making)}})
(marked,{{(marked)}})
(passed,{{(passed)}})
(should,{{(should)}})
(slowly,{{(slowly)}})
(sooner,{{(sooner)}})
(think,{{(think.)}})
(agreed.,{{(agreed.)}})
(mocking,{{(mocking)}})
(replied,{{(replied)}})
(runners,{{(runners)}})
(running,{{(running)}})
(started,{{(started)}})
(Tortoise,{{(Tortoise),(Tortoise),(Tortoise),(Tortoise),(Tortoise),(Tortoise),(Tortoise),(Tortoise),(Tortoise)})}
(distance,{{(distance)}})
(overtake,{{(overtake)}})
(steadily,{{(steadily)}})
(swiftest,{{(swiftest)}})
(anywhere?,{{(anywhere?)}})
(consented,{{(consented)}})
(meantime,{{(meantime)}})
(sleeping.,{{(sleeping.)}})
(ridiculous,{{(ridiculous)}})
(peacefully;,{(peacefully;)})
grunt> |
```

Comment : Group all the similar words together and store in groups variable

Step 4 : Count the number of words of each group

Command :

```
count_word= foreach groups generate group, (COUNT(words));
```

```
DUMP count_word;
```

Output :

```
hadoop@Lenovo: ~
Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime    MaxReduceTime   MinReduceT
ime     AvgReduceTime   MedianReducetime   Alias   Feature Outputs
job_local396631090_0002 1       1   n/a      n/a      n/a      n/a      n/a      n/a      n/a      count_word
,groups,lines,words      GROUP_BY,COMBINER           file:/tmp/temp-721289047/tmp-360633201,

Input(s):
Successfully read 11 records (3920 bytes) from: "hdfs://localhost:9000/hare.txt"

Output(s):
Successfully stored 116 records in: "file:/tmp/temp-721289047/tmp-360633201"

Counters:
Total records written : 116
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local396631090_0002

2022-04-25 01:45:13,810 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-04-25 01:45:13,813 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-04-25 01:45:13,815 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
```

```
hadoop@Lenovo: ~
2022-04-25 01:45:13,826 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input p
aths to process : 1
(&,1)
(A,1)
(I,1)
(a,7)
(Do,1)
(So,1)
(as,1)
(at,2)
(he,5)
(in,1)
(it,1)
(of,4)
(on,1)
(so,1)
(to,4)
(up,1)
(But,1)
(Fox,1)
(The,5)
(Yes,1)
(act,1)
(and,6)
(but,3)
(day,1)
(did,1)
(far,1)
(for,3)
(fun,2)
(get,2)
(his,1)
```

```
hadoop@Lenovo: ~
(him,1)
(his,1)
(how,1)
(it.,1)
(lay,1)
(nap,1)
(not,1)
(now,1)
(one,1)
(out,1)
(ran,1)
(run,1)
(the,19)
(try,1)
(up.,1)
(was,6)
(who,1)
(you,3)
(Hare,8)
(I'll,1)
(down,1)
(ever,1)
(feel,1)
(idea,1)
(kept,1)
(last,1)
(make,1)
(much,1)
(near,1)
(off.,1)
(race,3)
(his,1)
```

```
(near,1)
(off.,1)
(race,3)
(soon,1)
(take,1)
(than,1)
(time,1)
(very,2)
(wake,1)
(when,1)
(with,3)
(after,1)
(asked,1)
(being,1)
(catch,1)
(could,1)
(goal,1)
(going,1)
(judge,1)
(place,1)
(prove,1)
(sight,1)
(slept,1)
(slow.,1)
(there,1)
(thing,1)
(time.,1)
(until,1)
(where,1)
(amused,1)
(beside,1)
('----,1)
```

```
(where,1)
(amused,1)
(beside,1)
(course,1)
(deeply,1)
(laugh.,1)
(making,1)
(marked,1)
(passed,1)
(should,1)
(slowly,1)
(sooner,1)
(think.,1)
(agreed.,1)
(mocking,1)
(replied,1)
(runners,1)
(running,1)
(started,1)
(Tortoise,9)
(distance,1)
(overtake,1)
(steadily,1)
(swiftest,1)
(anywhere?,1)
(consented,1)
(meantime,1)
(sleeping.,1)
(ridiculous,1)
(peacefully;,1)
grunt> ■
```

Comment : Count the number of words of each groups which are similiar words and store in count_word variable, which will produce the final output of wordcount application.

Time_Series (Assignment-7)

Sumon Singh

2022-05-15

Load Library

```
library(timeSeries)

## Loading required package: timeDate

library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method           from
##   as.zoo.data.frame zoo
```

Read Dataset

```
df = read.csv('/home/sumon/Desktop/Python-Codes/Indrani-mam/Assignment-7/daily-total-female-births.csv')
head(df)
```

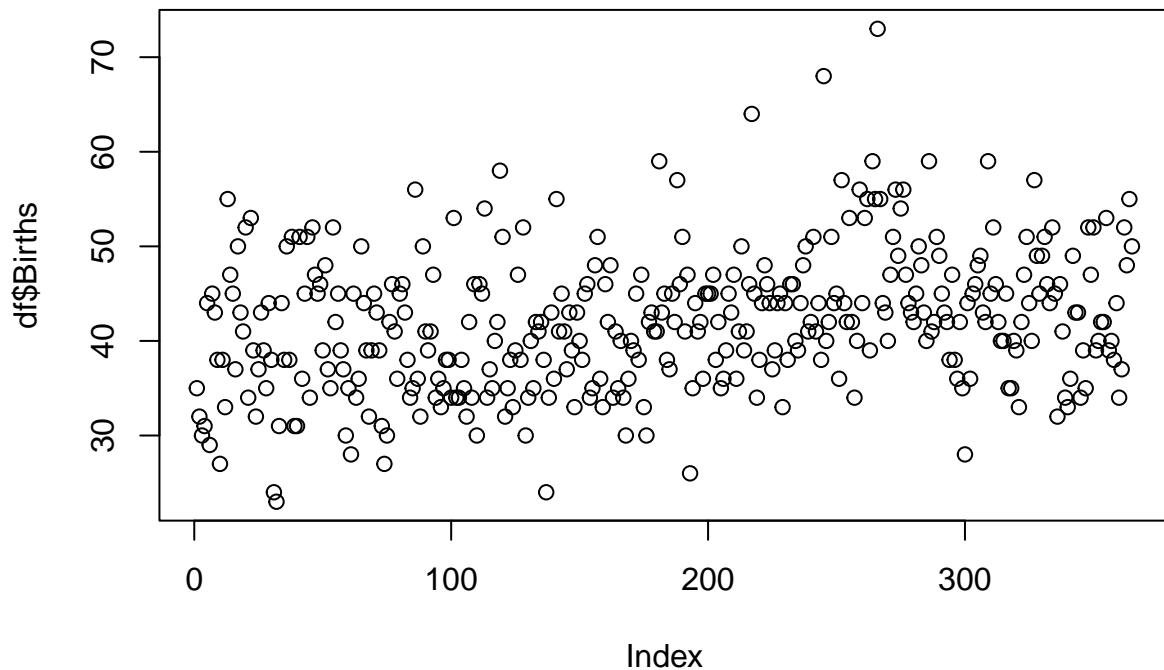
```
##          Date Births
## 1 1959-01-01     35
## 2 1959-01-02     32
## 3 1959-01-03     30
## 4 1959-01-04     31
## 5 1959-01-05     44
## 6 1959-01-06     29
```

Summary

```
summary(df)

##          Date              Births
##  Length:365        Min.    :23.00
##  Class :character  1st Qu.:37.00
##  Mode  :character  Median :42.00
##                  Mean   :41.98
##                  3rd Qu.:46.00
##                  Max.   :73.00
```

Plot



Data Transformation

```
row.names(df)=df$Date  
df_mod=df[,-c(1)]
```

Convert Dataset To TimeSeries

```
df_mod = ts(data = df_mod,frequency = 30)  
head(df_mod)
```

```
## Time Series:  
## Start = c(1, 1)  
## End = c(1, 6)  
## Frequency = 30  
## [1] 35 32 30 31 44 29
```

Print the start, end and frequency

```
start(df_mod)
```

```
## [1] 1 1
```

```
end(df_mod)
```

```
## [1] 13 5
```

```
frequency(df_mod)
```

```
## [1] 30
```

Decompose the dataset

```
decomposed_data = decompose(df_mod,"multiplicative")
decomposed_data
```

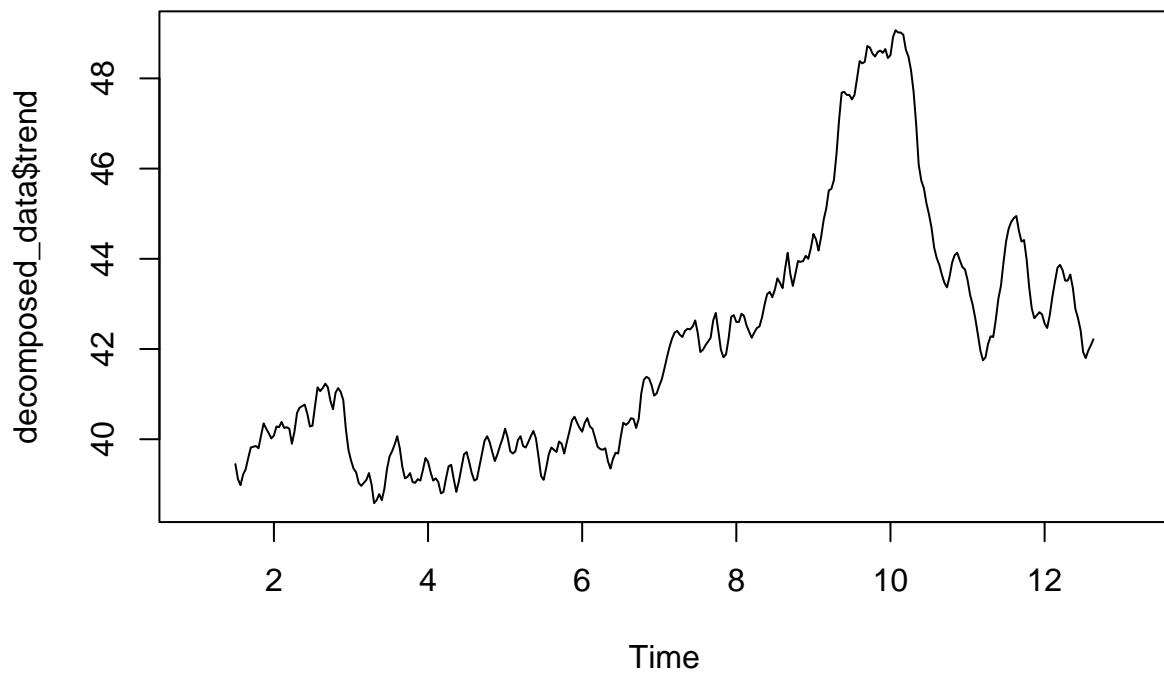
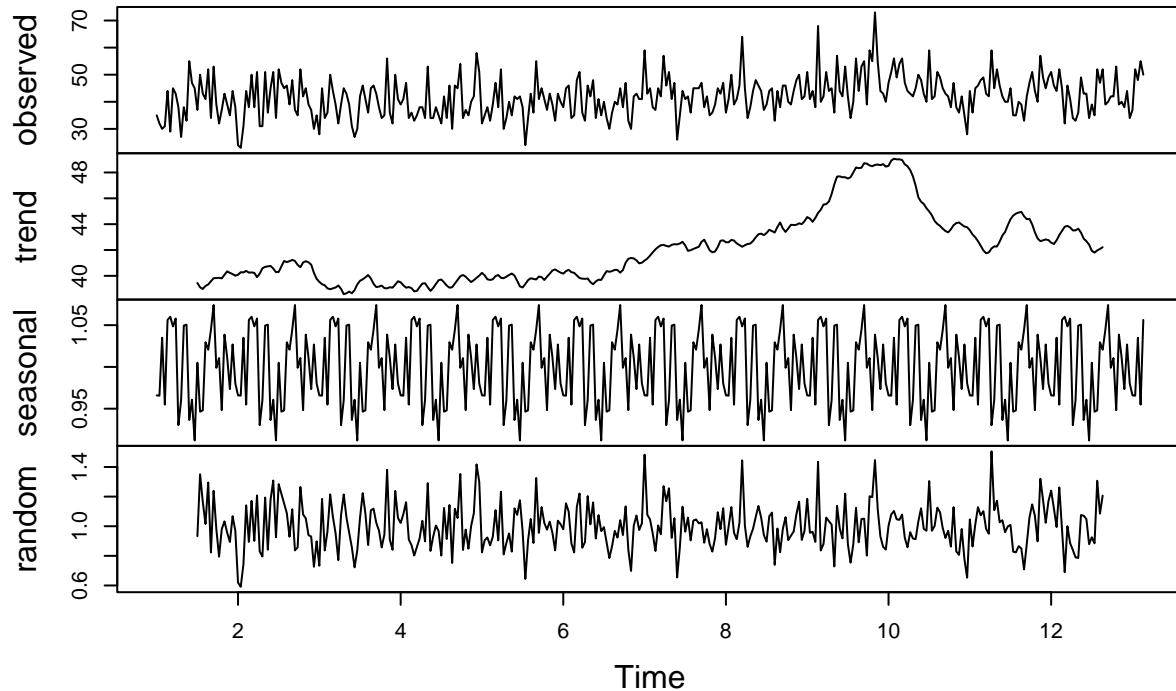
```
## $x
## Time Series:
## Start = c(1, 1)
## End = c(13, 5)
## Frequency = 30
##   [1] 35 32 30 31 44 29 45 43 38 27 38 33 55 47 45 37 50 43 41 52 34 53 39 32 37
##   [26] 43 39 35 44 38 24 23 31 44 38 50 38 51 31 31 51 36 45 51 34 52 47 45 46 39
##   [51] 48 37 35 52 42 45 39 37 30 35 28 45 34 36 50 44 39 32 39 45 43 39 31 27 30
##   [76] 42 46 41 36 45 46 43 38 34 35 56 36 32 50 41 39 41 47 34 36 33 35 38 38 34
##  [101] 53 34 34 38 35 32 42 34 46 30 46 45 54 34 37 35 40 42 58 51 32 35 38 33 39
##  [126] 47 38 52 30 34 40 35 42 41 42 38 24 34 43 36 55 41 45 41 37 43 39 33 43 40
##  [151] 38 45 46 34 35 48 51 36 33 46 42 48 34 41 35 40 34 30 36 40 39 45 38 47 33
##  [176] 30 42 43 41 41 59 43 45 38 37 45 42 57 46 51 41 47 26 35 44 41 42 36 45 45
##  [201] 45 47 38 42 35 36 39 45 43 47 36 41 50 39 41 46 64 45 34 38 44 48 46 44 37
##  [226] 39 44 45 33 44 38 46 46 40 39 44 48 50 41 42 51 41 44 38 68 40 42 51 44 45
##  [251] 36 57 44 42 53 42 34 40 56 44 53 55 39 59 55 73 55 44 43 40 47 51 56 49 54
##  [276] 56 47 44 43 42 45 50 48 43 40 59 41 42 51 49 45 43 42 38 47 38 36 42 35 28
##  [301] 44 36 45 46 48 49 43 42 59 45 52 46 42 40 40 45 35 35 40 39 33 42 47 51 44
##  [326] 40 57 49 45 49 51 46 44 52 45 32 46 41 34 33 36 49 43 43 34 39 35 52 47 52
##  [351] 39 40 42 42 53 39 40 38 44 34 37 52 48 55 50
##
## $seasonal
## Time Series:
## Start = c(1, 1)
## End = c(13, 5)
## Frequency = 30
##   [1] 0.9658815 0.9655662 1.0349163 0.9548504 1.0562436 1.0600917 1.0485235
##   [8] 1.0579387 0.9302591 0.9603238 1.0494971 1.0506181 0.9363863 0.9607381
```

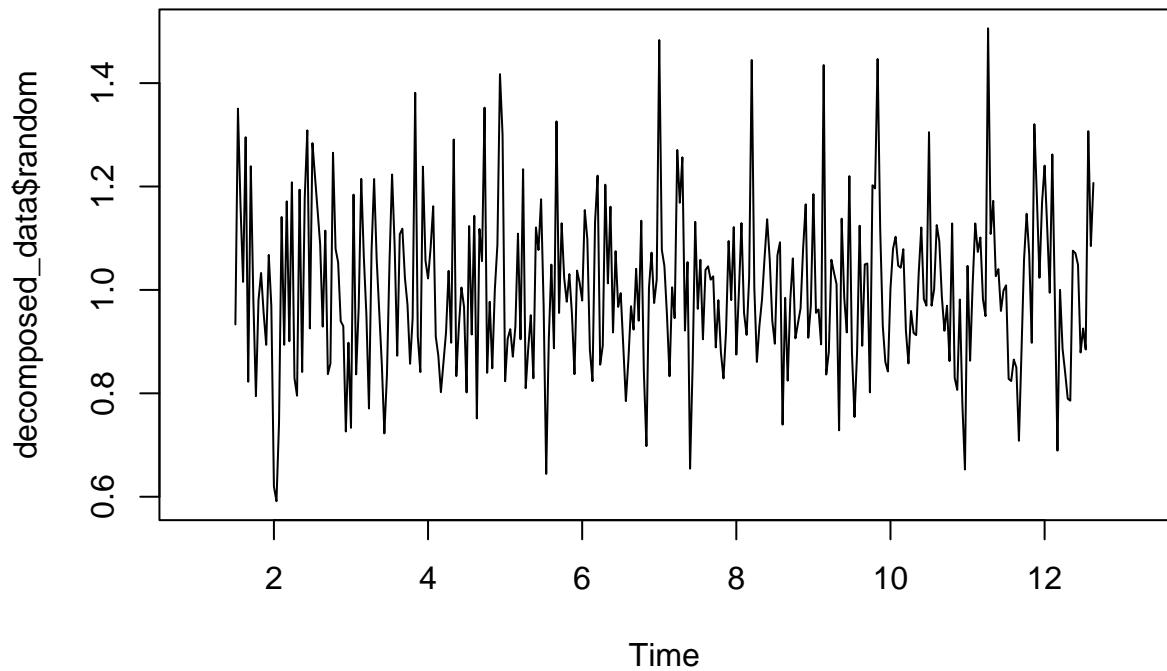


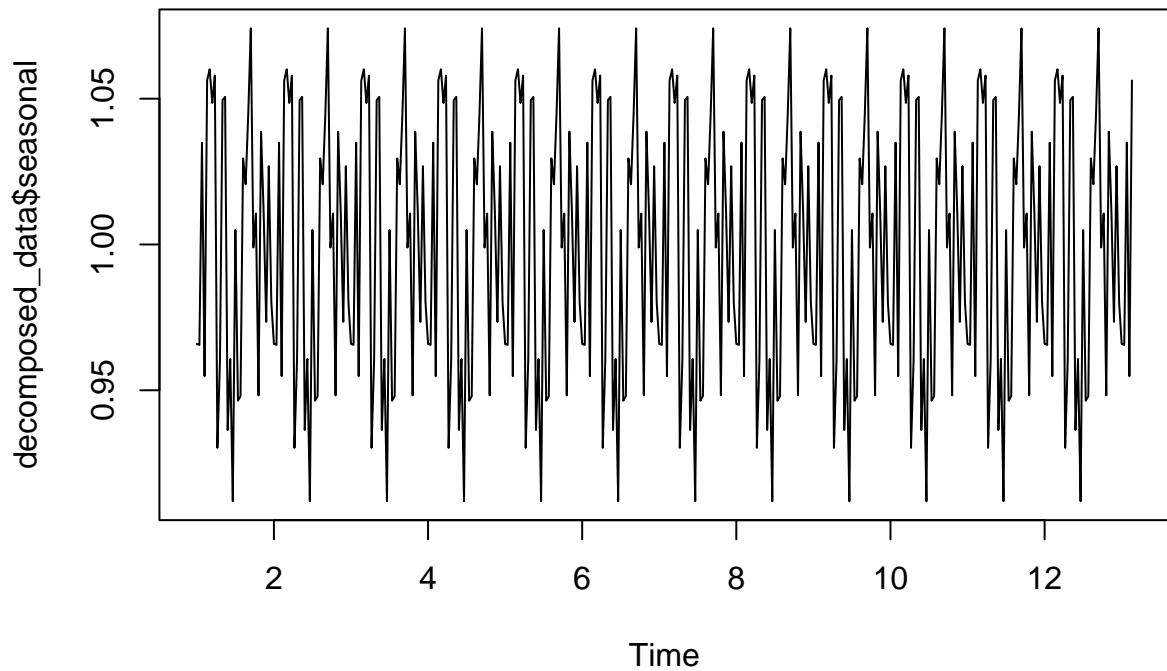
```
##  
## $figure  
## [1] 0.9658815 0.9655662 1.0349163 0.9548504 1.0562436 1.0600917 1.0485235  
## [8] 1.0579387 0.9302591 0.9603238 1.0494971 1.0506181 0.9363863 0.9607381  
## [15] 0.9119394 1.0049725 0.9464093 0.9479546 1.0295201 1.0206300 1.0443615  
## [22] 1.0741232 0.9989503 1.0106242 0.9482386 1.0386889 1.0112643 0.9734342  
## [29] 1.0268625 0.9801920  
##  
## $type  
## [1] "multiplicative"  
##  
## attr(),"class")  
## [1] "decomposed.ts"
```

Explore its components

Decomposition of multiplicative time series



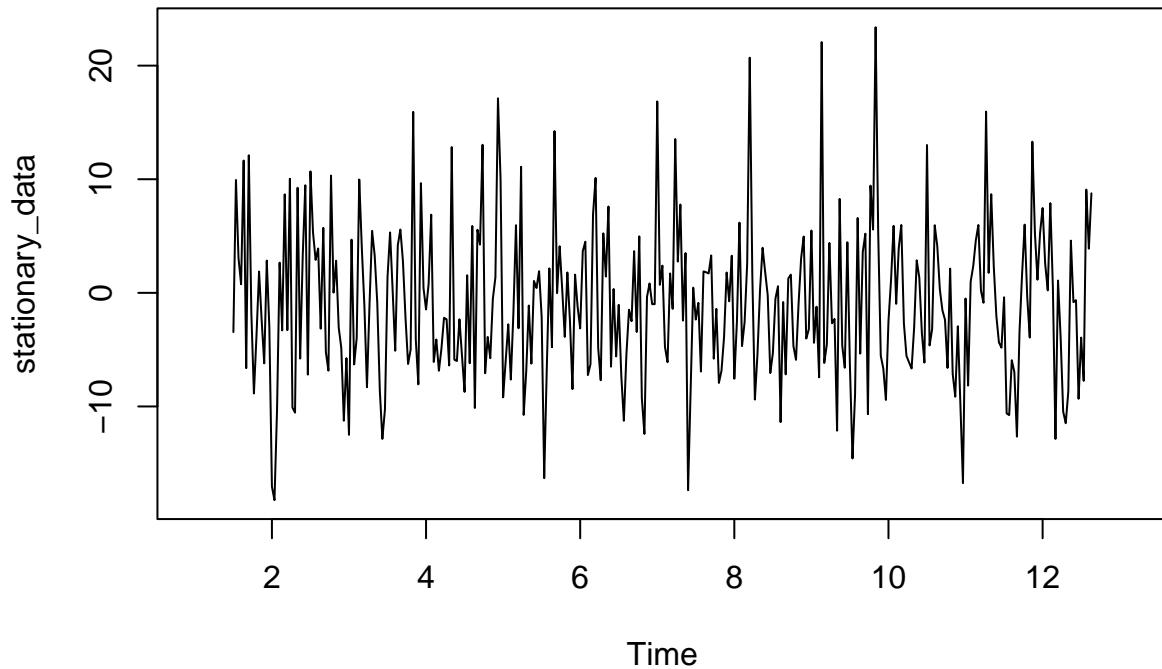




Remove the seasonality and trend

```
stationary_data=df_mod-decomposed_data$seasonal-decomposed_data$trend
```

Plot stationary_data



Apply ARIMA

```
f_model=auto.arima(stationary_data)
f_model

## Series: stationary_data
## ARIMA(1,0,0)(1,0,0) [30] with non-zero mean
##
## Coefficients:
##             ar1      sar1      mean
##             0.0965  -0.0264  -0.9689
## s.e.    0.0081   0.0038   0.4005
## 
## sigma^2 = 46.44: log likelihood = -1116.72
## AIC=2241.45   AICc=2241.57   BIC=2256.7
```

Forecast the number of female births for the next 5 months

```
f=forecast(f_model,h=5,level=c(95))  
f
```

```
##           Point Forecast      Lo 95     Hi 95  
## 13.16667    -0.6546930 -14.07381 12.76443  
## 13.20000    -1.0231710 -14.44229 12.39595  
## 13.23333    -0.8938857 -14.31300 12.52523  
## 13.26667    -0.7184581 -14.13758 12.70066  
## 13.30000    -0.6912399 -14.11036 12.72788
```

plot the forecast

Forecasts from ARIMA(1,0,0)(1,0,0)[30] with non-zero mean

