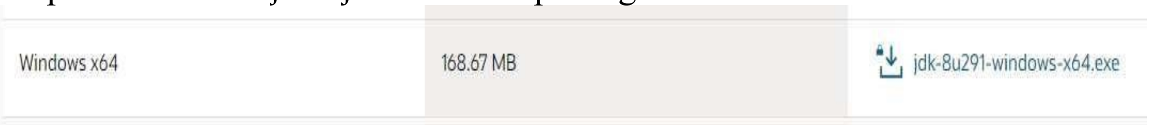
PRACTICAL NO : 1

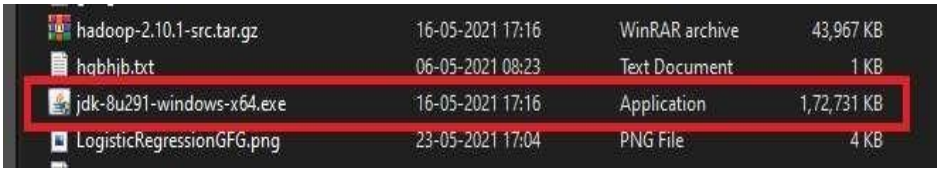
Aim: Install, configure and run Hadoop and HDFS

Hadoop Installation.

Step 1: downlaod java jdk first .the package size 168.67MB

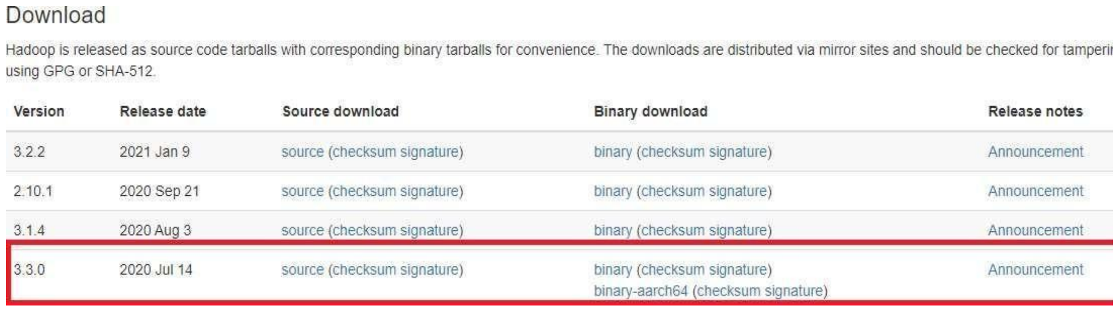


Windows x64	168.67 MB	jdk-8u291-windows-x64.exe



hadoop-2.10.1-src.tar.gz	16-05-2021 17:16	WinRAR archive	43,967 KB
hgbhjb.txt	06-05-2021 08:23	Text Document	1 KB
jdk-8u291-windows-x64.exe	16-05-2021 17:16	Application	1,72,731 KB
LogisticRegressionGFG.png	23-05-2021 17:04	PNG File	4 KB

Step 2: download Hadoop binaries from the official website. The binary package size is about 342 MB.

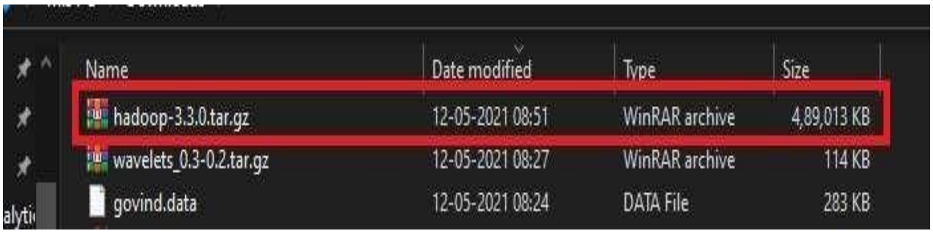


Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tamperi using GPG or SHA-512.

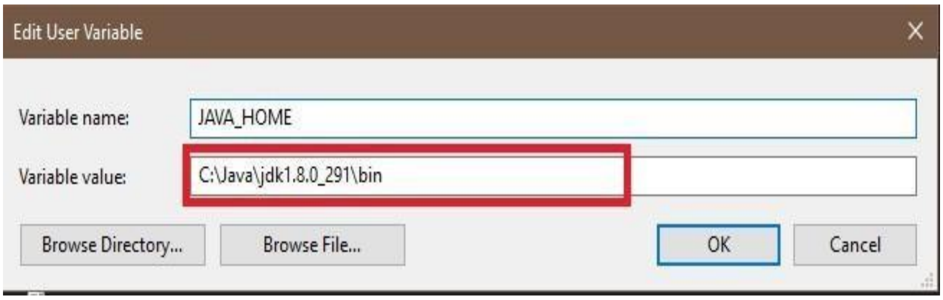
Version	Release date	Source download	Binary download	Release notes
3.2.2	2021 Jan 9	source (checksum signature)	binary (checksum signature)	Announcement
2.10.1	2020 Sep 21	source (checksum signature)	binary (checksum signature)	Announcement
3.1.4	2020 Aug 3	source (checksum signature)	binary (checksum signature)	Announcement
3.3.0	2020 Jul 14	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement

Step 3: After finishing the file download, we should unpack the package using 7zip inttwo steps. First, we should extract the hadoop -3.2.1 .tar.gz library, and then, we shouldunpack the extracted tar file:



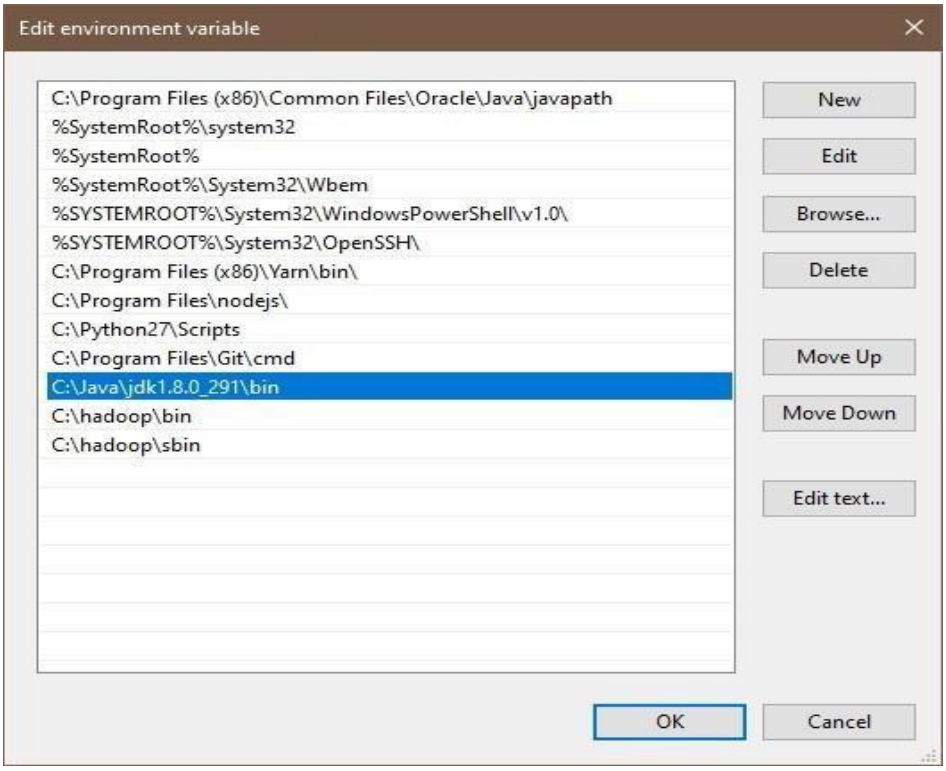
Name	Date modified	Type	Size
hadoop-3.3.0.tar.gz	12-05-2021 08:51	WinRAR archive	4,89,013 KB
wavelets_0.3-0.2.tar.gz	12-05-2021 08:27	WinRAR archive	114 KB
govind.data	12-05-2021 08:24	DATA File	283 KB

Step 4: When the “Advanced system settings” dialog appears, go to the “Advanced” tab and click on the “Environment variables” button located on the bottom of the dialog.



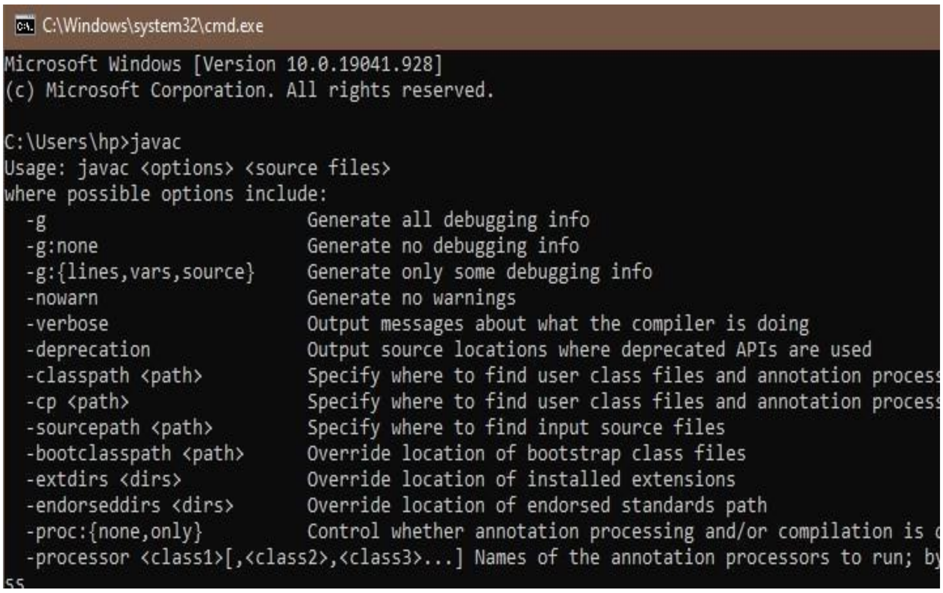
Variable name: JAVA_HOME

Variable value: C:\Java\jdk1.8.0_291\bin



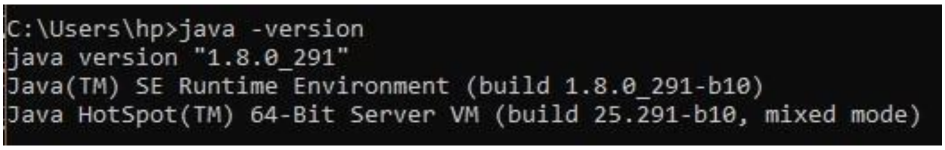
```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\Yarn\bin\
C:\Program Files\nodejs\
C:\Python27\Scripts
C:\Program Files\Git\cmd
C:\Java\jdk1.8.0_291\bin
C:\hadoop\bin
C:\hadoop\sbin
```

Step 5: Check the version of java



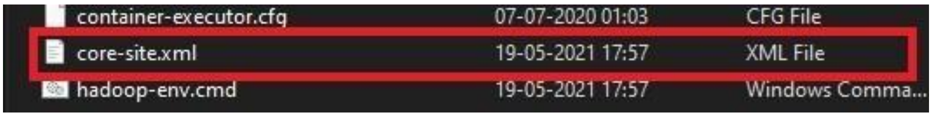
C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>javac
Usage: javac <options> <source files>
where possible options include:
  -g                         Generate all debugging info
  -g:none                    Generate no debugging info
  -g:{lines,vars,source}     Generate only some debugging info
  -nowarn                    Generate no warnings
  -verbose                   Output messages about what the compiler is doing
  -deprecation               Output source locations where deprecated APIs are used
  -classpath <path>          Specify where to find user class files and annotation process
  -cp <path>                 Specify where to find user class files and annotation process
  -sourcepath <path>         Specify where to find input source files
  -bootclasspath <path>      Override location of bootstrap class files
  -extdirs <dirs>            Override location of installed extensions
  -endorseddirs <dirs>       Override location of endorsed standards path
  -proc:{none,only}          Control whether annotation processing and/or compilation is d
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; by
ss
```

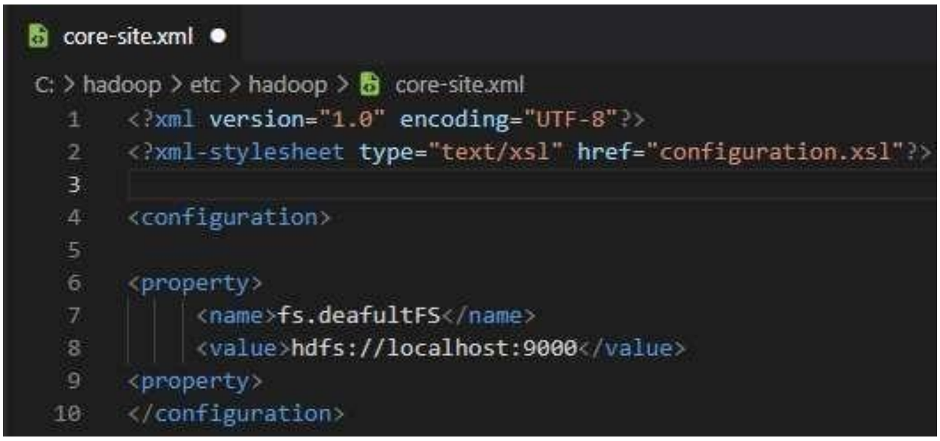


```
C:\Users\hp>java -version
java version "1.8.0_291"
Java(TM) SE Runtime Environment (build 1.8.0_291-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.291-b10, mixed mode)
```

Step 6: Configuration core-site.xml

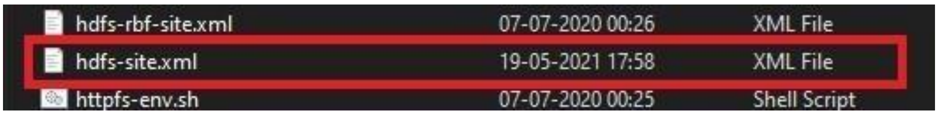


```
container-executor.cfg      07-07-2020 01:03      CFG File
core-site.xml               19-05-2021 17:57      XML File
hadoop-env.cmd              19-05-2021 17:57      Windows Comma...
```

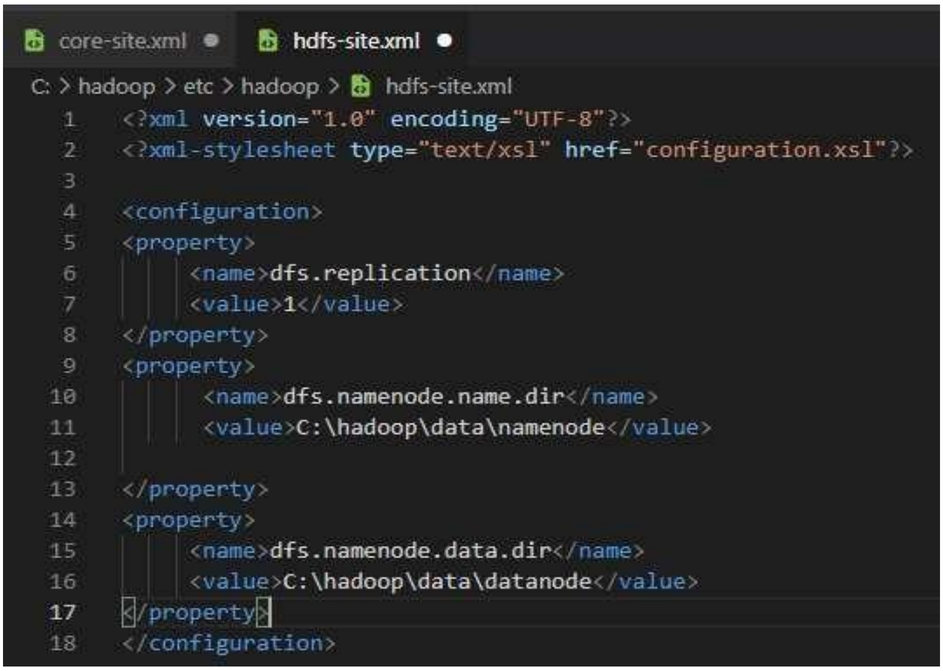


```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>
        <name>fs.deafultFS</name>
        <value>hdfs://localhost:9000</value>
<property>
</configuration>
```

Step 7: Configuration core-site.xml

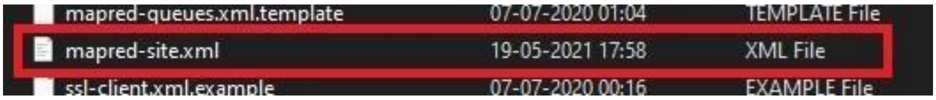


```
hdfs-rbf-site.xml     07-07-2020 00:26     XML File
hdfs-site.xml         19-05-2021 17:58     XML File
httpfs-env.sh         07-07-2020 00:25     Shell Script
```

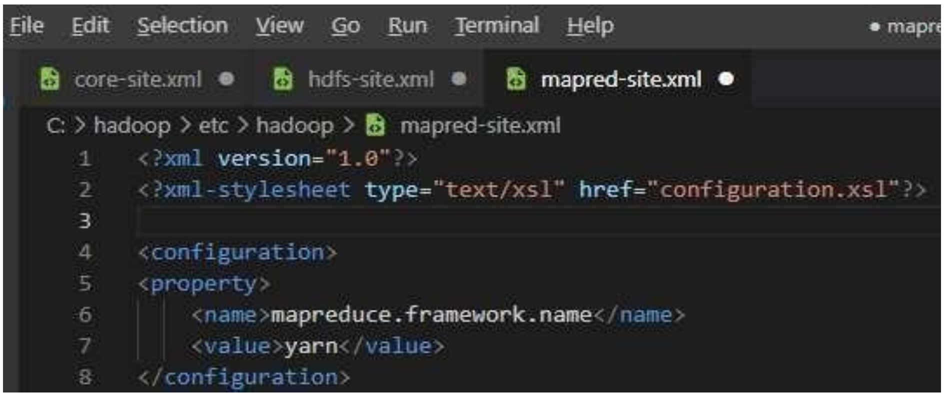


```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
<property>
        <name>dfs.replication</name>
        <value>1</value>
</property>
<property>
          <name>dfs.namenode.name.dir</name>
          <value>C:\hadoop\data\namenode</value>

</property>
<property>
        <name>dfs.namenode.data.dir</name>
        <value>C:\hadoop\data\datanode</value>
</property>
</configuration>
```

Step 8: Configuration core-site.xml



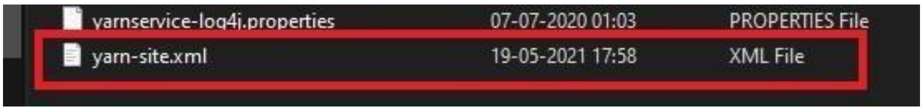
```
mapred-queues.xml.template     07-07-2020 01:04     TEMPLATE File
mapred-site.xml                19-05-2021 17:58     XML File
ssl-client.xml.example         07-07-2020 00:16     EXAMPLE File
```



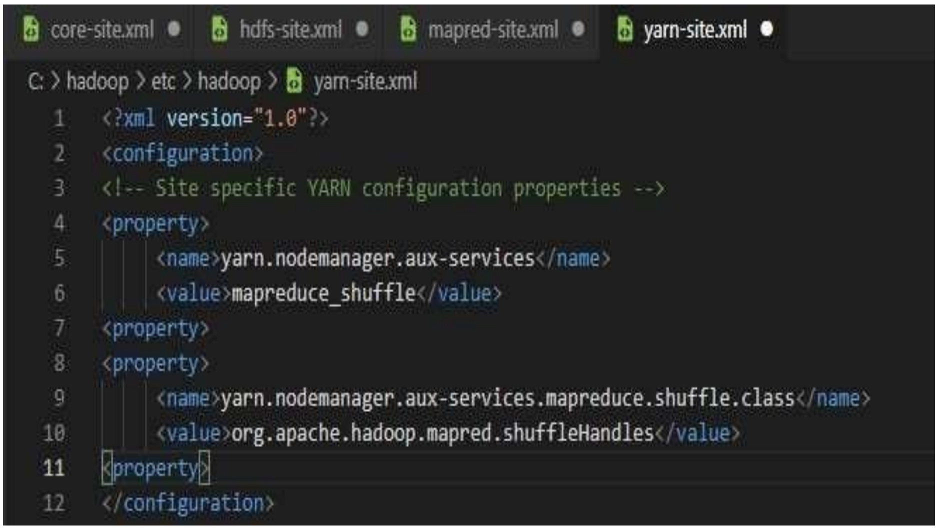
File Edit Selection View Go Run Terminal Help
● mapre
core-site.xml ●
hdfs-site.xml ●
mapred-site.xml ●
C: > hadoop > etc > hadoop > mapred-site.xml

```xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</configuration>
```

Step 9: Configuration core-site.xml

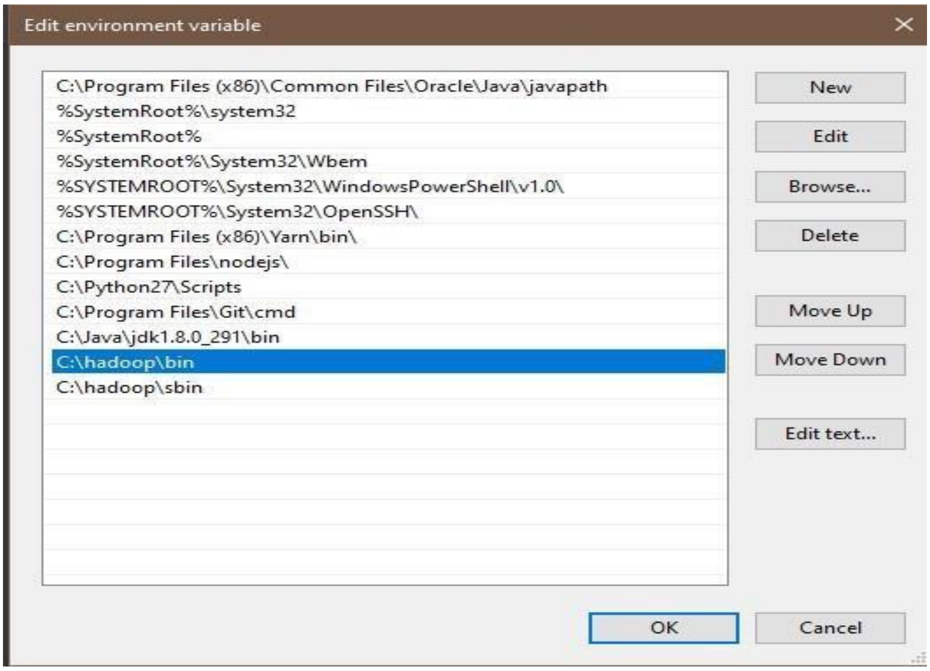


```
yarnservice-log4j.properties     07-07-2020 01:03     PROPERTIES File
yarn-site.xml                    19-05-2021 17:58     XML File
```

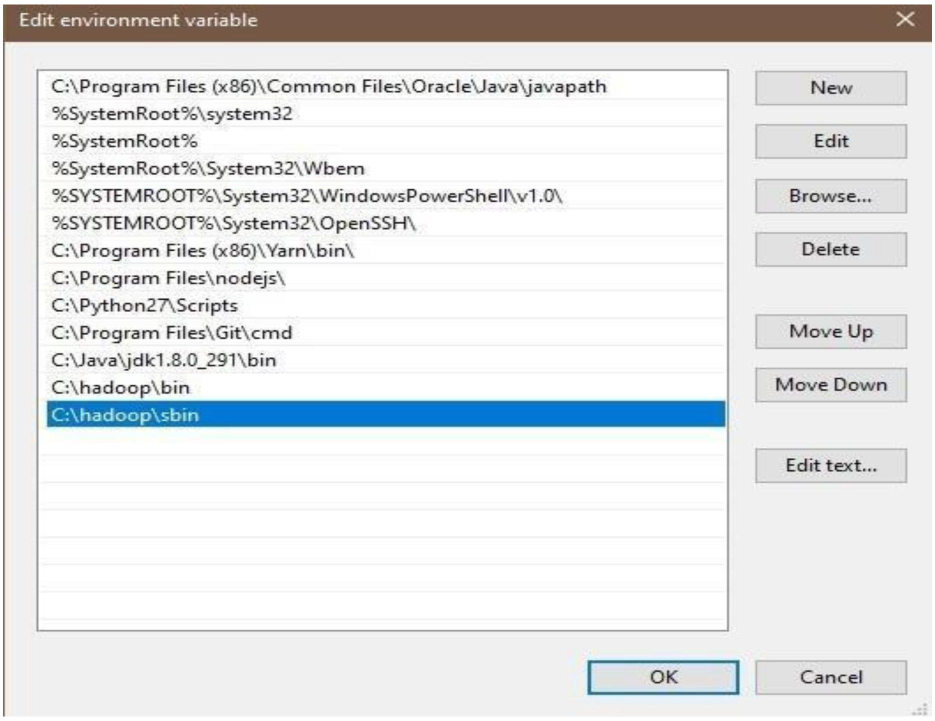


```
<?xml version="1.0"?>
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
<property>
<property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.shuffleHandles</value>
<property>
</configuration>
```

Step 10: When the “Advanced system settings” dialog appears, go to the “Advanced” tab and click on the “Environment variables” button located on the bottom of the dialog.



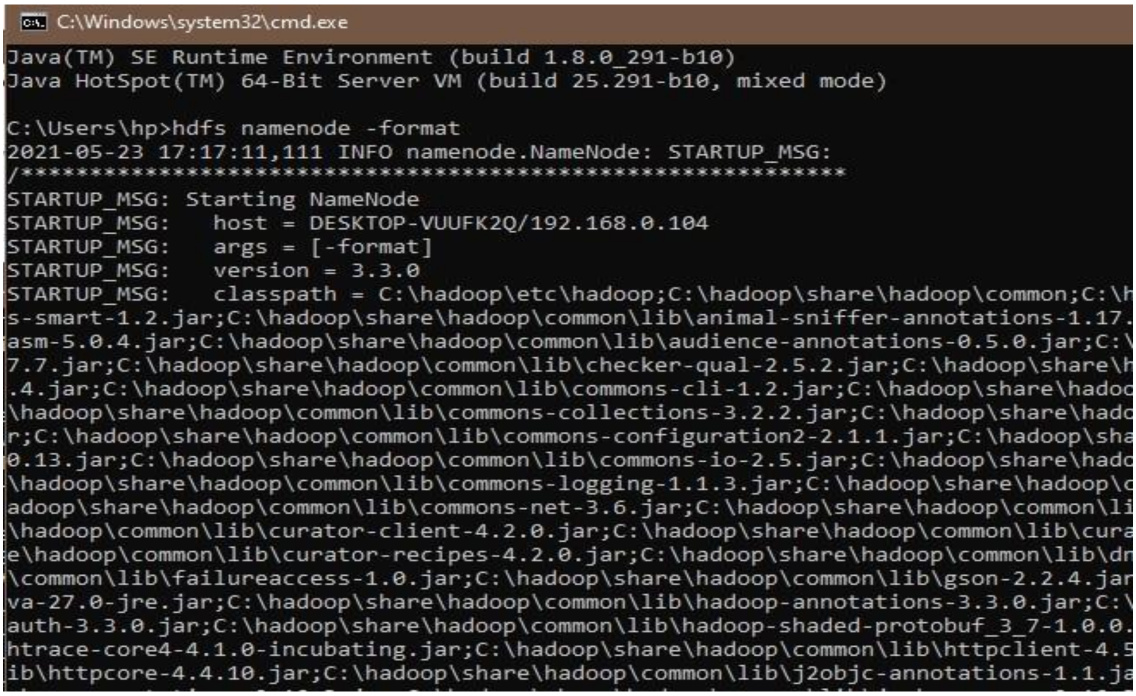
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\Yarn\bin\
C:\Program Files\nodejs\
C:\Python27\Scripts
C:\Program Files\Git\cmd
C:\Java\jdk1.8.0_291\bin
C:\hadoop\bin
C:\hadoop\sbin



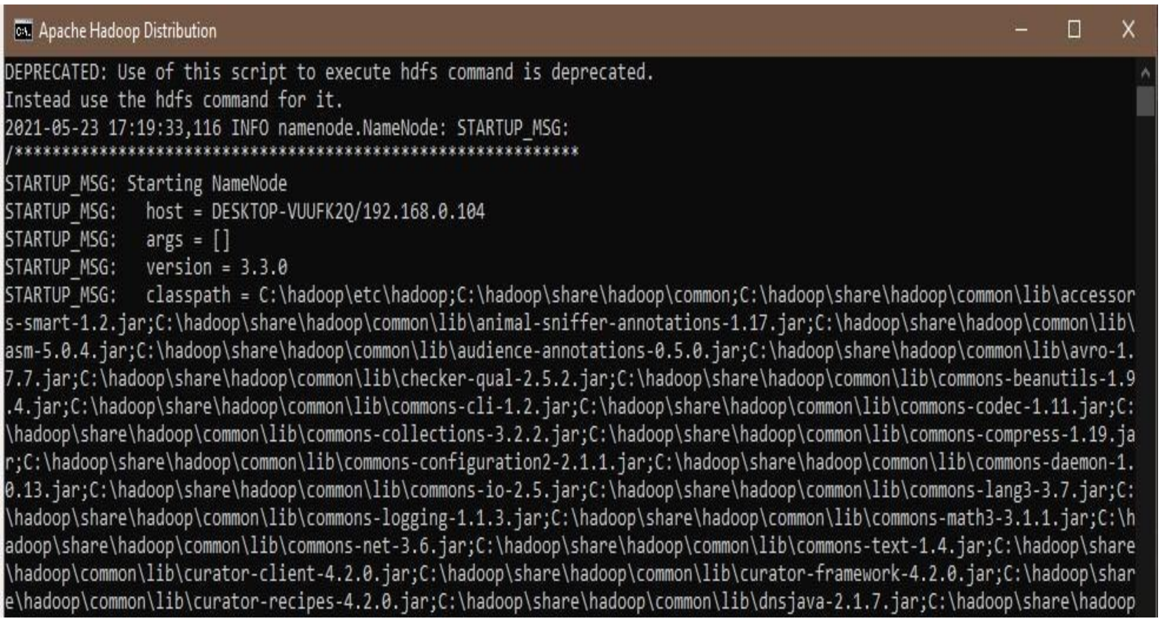
Edit environment variable

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\Yarn\bin\
C:\Program Files\nodejs\
C:\Python27\Scripts
C:\Program Files\Git\cmd
C:\Java\jdk1.8.0_291\bin
C:\hadoop\bin
C:\hadoop\sbin
```

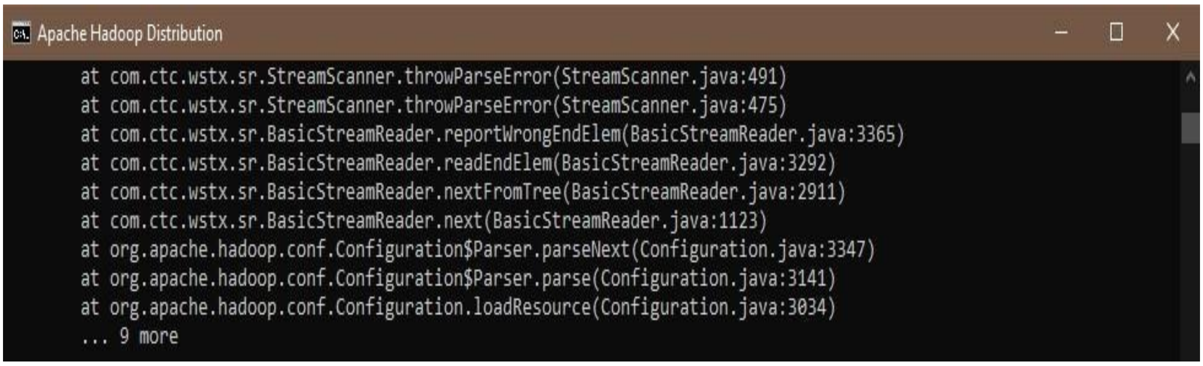
Step 11: let's check Hadoop install Successfully


C:\Windows\system32\cmd.exe

```
Java(TM) SE Runtime Environment (build 1.8.0_291-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.291-b10, mixed mode)

C:\Users\hp>hdfs namenode -format
2021-05-23 17:17:11,111 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-VUUFK2Q/192.168.0.104
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = C:\hadoop\etc\hadoop;C:\hadoop\share\hadoop\common;C:\h
s-smart-1.2.jar;C:\hadoop\share\hadoop\common\lib\animal-sniffer-annotations-1.17.
asm-5.0.4.jar;C:\hadoop\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\
7.7.jar;C:\hadoop\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop\share\h
.4.jar;C:\hadoop\share\hadoop\common\lib\commons-cli-1.2.jar;C:\hadoop\share\hadoo
\hadoop\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop\share\hado
r;C:\hadoop\share\hadoop\common\lib\commons-configuration2-2.1.1.jar;C:\hadoop\sha
0.13.jar;C:\hadoop\share\hadoop\common\lib\commons-io-2.5.jar;C:\hadoop\share\hado
\hadoop\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop\share\hadoop\c
adoop\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop\share\hadoop\common\li
\hadoop\common\lib\curator-client-4.2.0.jar;C:\hadoop\share\hadoop\common\lib\cura
e\hadoop\common\lib\curator-recipes-4.2.0.jar;C:\hadoop\share\hadoop\common\lib\dn
\common\lib\failureaccess-1.0.jar;C:\hadoop\share\hadoop\common\lib\gson-2.2.4.jar
va-27.0-jre.jar;C:\hadoop\share\hadoop\common\lib\hadoop-annotations-3.3.0.jar;C:\
auth-3.3.0.jar;C:\hadoop\share\hadoop\common\lib\hadoop-shaded-protobuf_3_7-1.0.0.
htrace-core4-4.1.0-incubating.jar;C:\hadoop\share\hadoop\common\lib\httpclient-4.5
ib\httpcore-4.4.10.jar;C:\hadoop\share\hadoop\common\lib\j2objc-annotations-1.1.ja
```

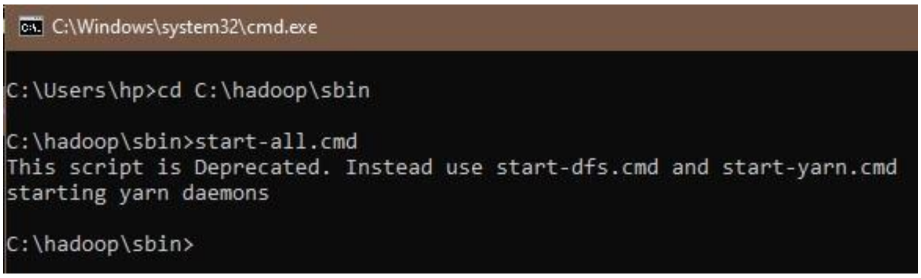


```
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
2021-05-23 17:19:33,116 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-VUUFK2Q/192.168.0.104
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.3.0
STARTUP_MSG:   classpath = C:\hadoop\etc\hadoop;C:\hadoop\share\hadoop\common;C:\hadoop\share\hadoop\common\lib\accessor
s-smart-1.2.jar;C:\hadoop\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop\share\hadoop\common\lib\
asm-5.0.4.jar;C:\hadoop\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\hadoop\share\hadoop\common\lib\avro-1.
7.7.jar;C:\hadoop\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop\share\hadoop\common\lib\commons-beanutils-1.9
.4.jar;C:\hadoop\share\hadoop\common\lib\commons-cli-1.2.jar;C:\hadoop\share\hadoop\common\lib\commons-codec-1.11.jar;C:
\hadoop\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop\share\hadoop\common\lib\commons-compress-1.19.ja
r;C:\hadoop\share\hadoop\common\lib\commons-configuration2-2.1.1.jar;C:\hadoop\share\hadoop\common\lib\commons-daemon-1.
0.13.jar;C:\hadoop\share\hadoop\common\lib\commons-io-2.5.jar;C:\hadoop\share\hadoop\common\lib\commons-lang3-3.7.jar;C:
\hadoop\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop\share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\h
adoop\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop\share
\hadoop\common\lib\curator-client-4.2.0.jar;C:\hadoop\share\hadoop\common\lib\curator-framework-4.2.0.jar;C:\hadoop\shar
e\hadoop\common\lib\curator-recipes-4.2.0.jar;C:\hadoop\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop\share\hadoop
```



```
        at com.ctc.wstx.sr.StreamScanner.throwParseError(StreamScanner.java:491)
        at com.ctc.wstx.sr.StreamScanner.throwParseError(StreamScanner.java:475)
        at com.ctc.wstx.sr.BasicStreamReader.reportWrongEndElem(BasicStreamReader.java:3365)
        at com.ctc.wstx.sr.BasicStreamReader.readEndElem(BasicStreamReader.java:3292)
        at com.ctc.wstx.sr.BasicStreamReader.nextFromTree(BasicStreamReader.java:2911)
        at com.ctc.wstx.sr.BasicStreamReader.next(BasicStreamReader.java:1123)
        at org.apache.hadoop.conf.Configuration$Parser.parseNext(Configuration.java:3347)
        at org.apache.hadoop.conf.Configuration$Parser.parse(Configuration.java:3141)
        at org.apache.hadoop.conf.Configuration.loadResource(Configuration.java:3034)
        ... 9 more
```

Step 12: Let check bin

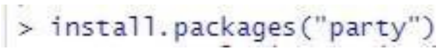


```
C:\Users\hp>cd C:\hadoop\sbin

C:\hadoop\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop\sbin>
```

PRACTICAL NO : 2

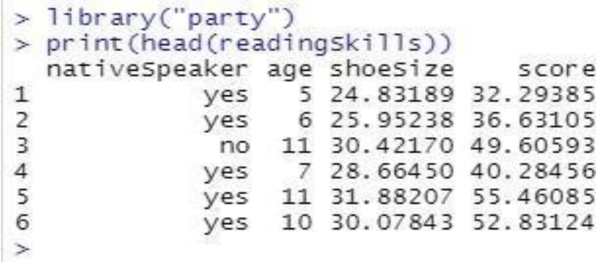
Aim: Implement Decision tree classification techniques.

Step 1: The package "party" has the function ctree() which is used to create andanalyze decison tree.



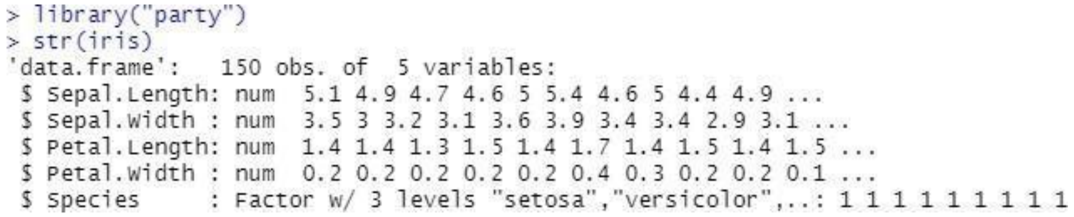
```
> install.packages("party")
```

Step 2: Load the party package. It will automatically load other# dependent packagesPrint some records from data set readingSkills.

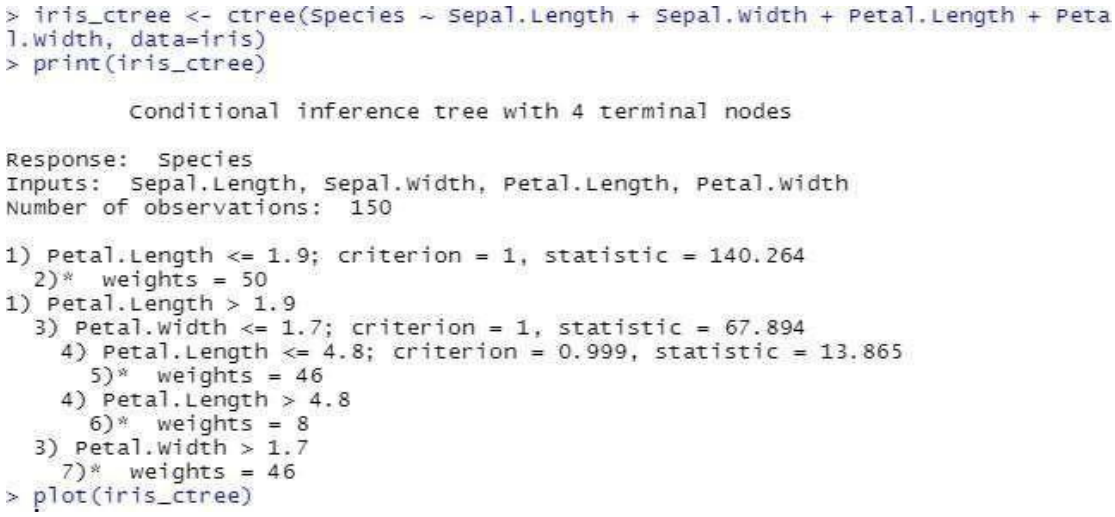


```
> library("party")
> print(head(readingSkills))
  nativeSpeaker age shoeSize    score
1           yes   5 24.83189 32.29385
2           yes   6 25.95238 36.63105
3            no  11 30.42170 49.60593
4           yes   7 28.66450 40.28456
5           yes  11 31.88207 55.46085
6           yes  10 30.07843 52.83124
>
```

Step 3 : Call function ctree to build a decision tree. The first parameter is a formula,which defines a target variable and a list of independent variables.

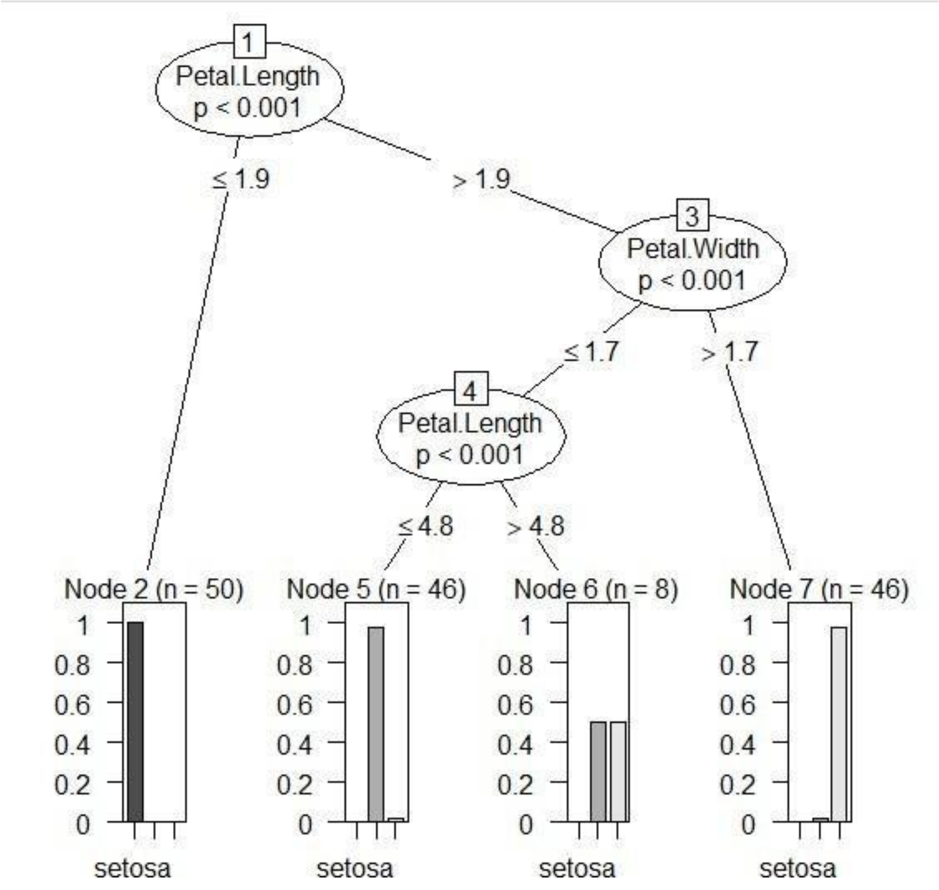


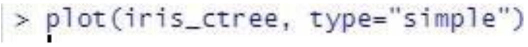
```
> library("party")
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1
```



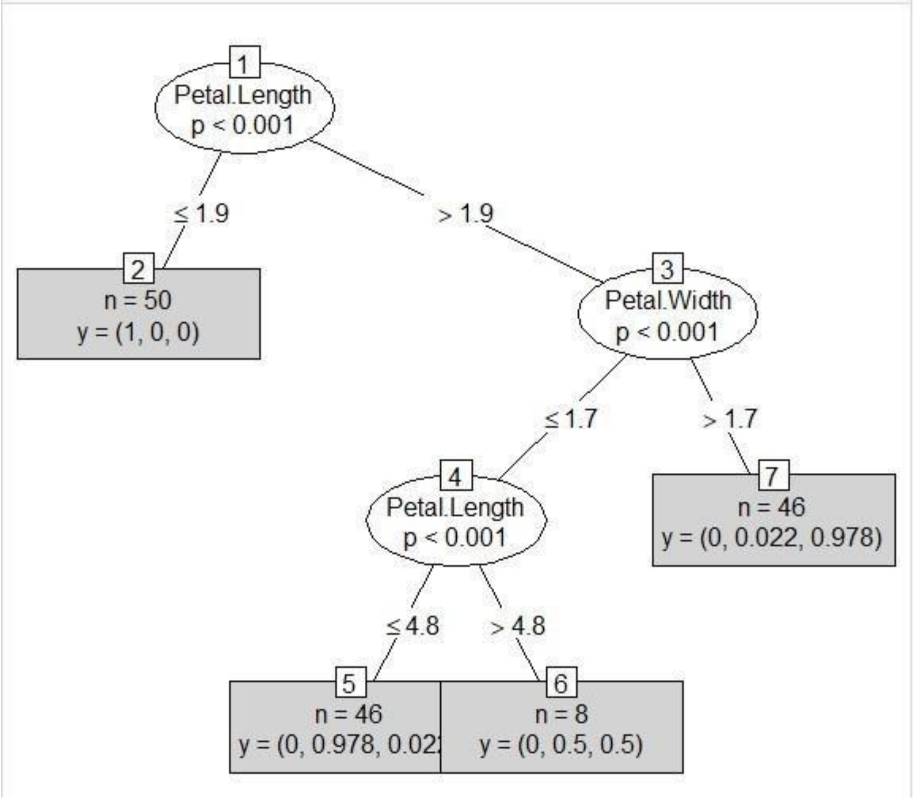
```
> iris_ctree <- ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Peta
l.Width, data=iris)
> print(iris_ctree)

         Conditional inference tree with 4 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  150

1) Petal.Length <= 1.9; criterion = 1, statistic = 140.264
  2)*  weights = 50
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 67.894
    4) Petal.Length <= 4.8; criterion = 0.999, statistic = 13.865
      5)*  weights = 46
    4) Petal.Length > 4.8
      6)*  weights = 8
  3) Petal.Width > 1.7
    7)*  weights = 46
> plot(iris_ctree)
```

Output :





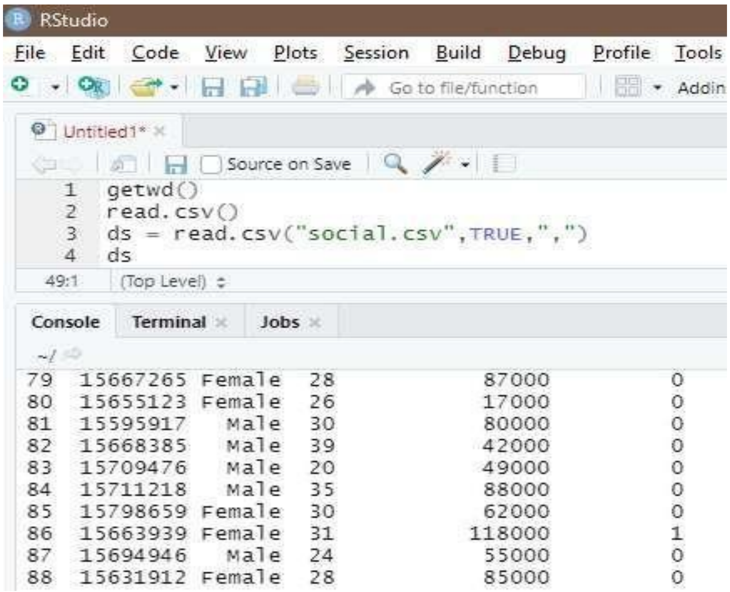
```
> plot(iris_ctree, type="simple")
```


1
Petal.Length
p < 0.001
≤ 1.9
> 1.9
2
n = 50
y = (1, 0, 0)
3
Petal.Width
p < 0.001
≤ 1.7
> 1.7
4
Petal.Length
p < 0.001
7
n = 46
y = (0, 0.022, 0.978)
≤ 4.8
> 4.8
5
n = 46
y = (0, 0.978, 0.02
6
n = 8
y = (0, 0.5, 0.5)

PRACTICAL NO : 3

Aim: Classification using SVM

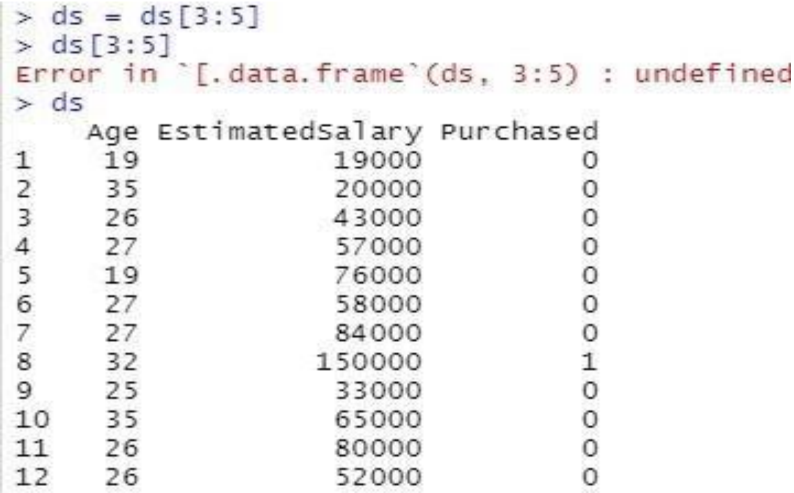
Step 1: Importing the dataset



```
getwd()
read.csv()
ds = read.csv("social.csv",TRUE,",")
ds
```

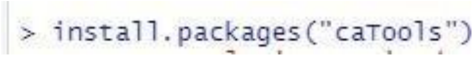
```
79  15667265 Female  28          87000         0
80  15655123 Female  26          17000         0
81  15595917   Male  30          80000         0
82  15668385   Male  39          42000         0
83  15709476   Male  20          49000         0
84  15711218   Male  35          88000         0
85  15798659 Female  30          62000         0
86  15663939 Female  31         118000         1
87  15694946   Male  24          55000         0
88  15631912 Female  28          85000         0
```

Step 2: Selecting columns 3-5



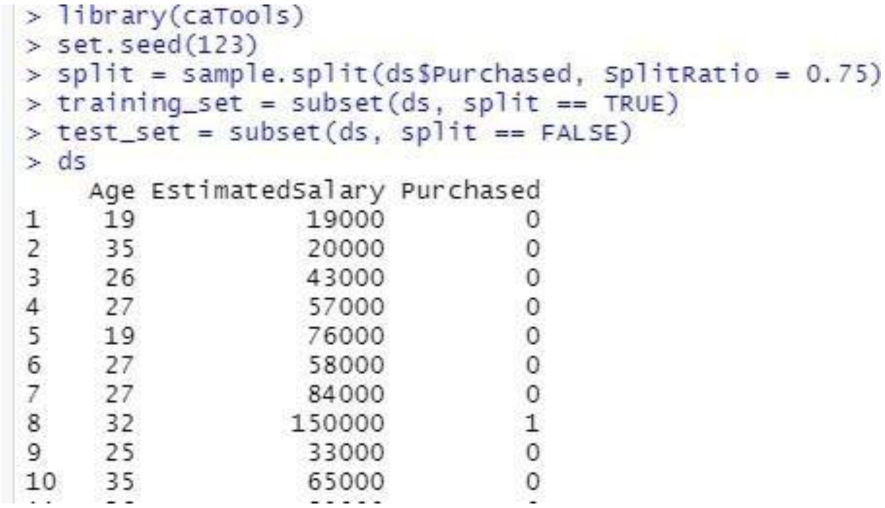
```
> ds = ds[3:5]
> ds[3:5]
Error in `[.data.frame`(ds, 3:5) : undefined
> ds
   Age EstimatedSalary Purchased
1   19           19000         0
2   35           20000         0
3   26           43000         0
4   27           57000         0
5   19           76000         0
6   27           58000         0
7   27           84000         0
8   32          150000         1
9   25           33000         0
10  35           65000         0
11  26           80000         0
12  26           52000         0
```

Step 3: install package



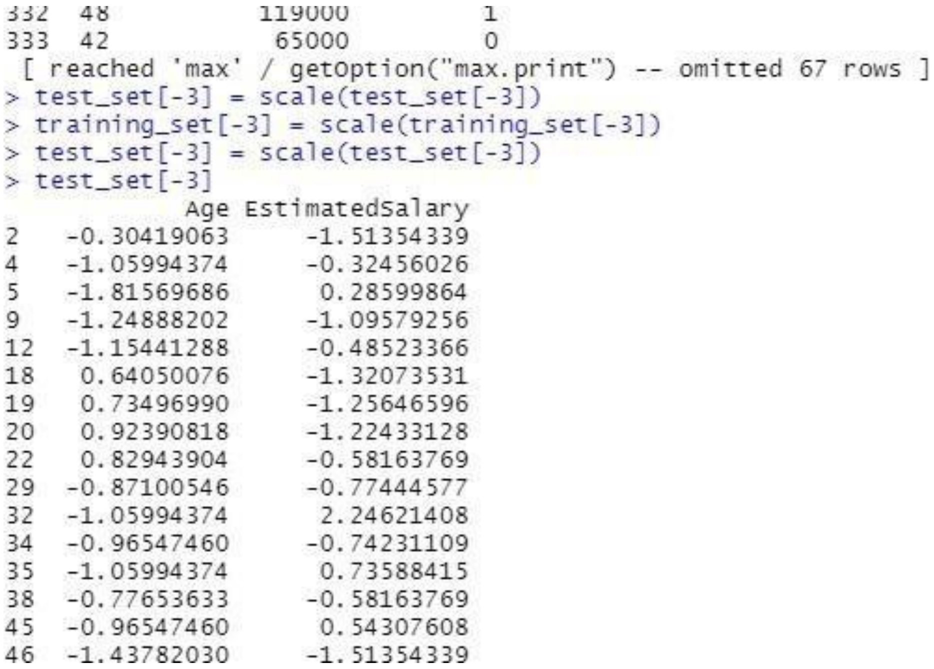
```
> install.packages("caTools")
```

Step 4: Splitting the dataset



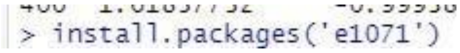
```
> library(caTools)
> set.seed(123)
> split = sample.split(ds$Purchased, SplitRatio = 0.75)
> training_set = subset(ds, split == TRUE)
> test_set = subset(ds, split == FALSE)
> ds
    Age EstimatedSalary Purchased
1    19           19000         0
2    35           20000         0
3    26           43000         0
4    27           57000         0
5    19           76000         0
6    27           58000         0
7    27           84000         0
8    32          150000         1
9    25           33000         0
10   35           65000         0
```

Step 5: Feature Scaling

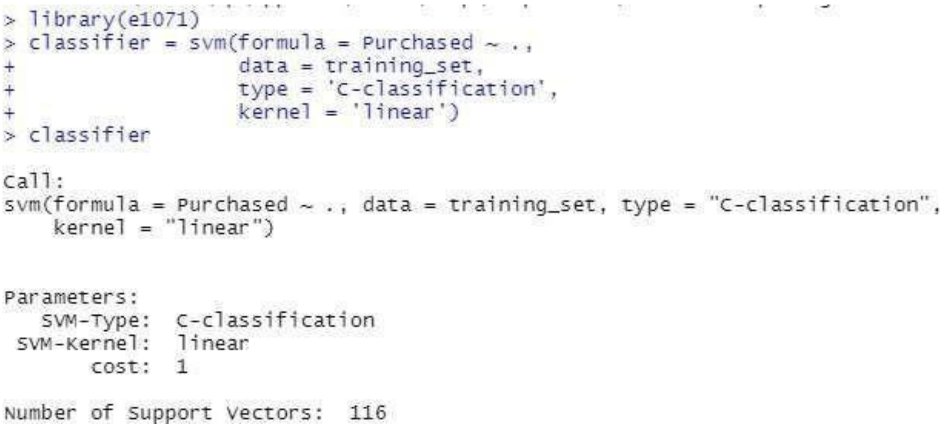


```
332  48          119000         1
333  42           65000         0
 [ reached 'max' / getOption("max.print") -- omitted 67 rows ]
> test_set[-3] = scale(test_set[-3])
> training_set[-3] = scale(training_set[-3])
> test_set[-3] = scale(test_set[-3])
> test_set[-3]
           Age EstimatedSalary
2   -0.30419063     -1.51354339
4   -1.05994374     -0.32456026
5   -1.81569686      0.28599864
9   -1.24888202     -1.09579256
12  -1.15441288     -0.48523366
18   0.64050076     -1.32073531
19   0.73496990     -1.25646596
20   0.92390818     -1.22433128
22   0.82943904     -0.58163769
29  -0.87100546     -0.77444577
32  -1.05994374      2.24621408
34  -0.96547460     -0.74231109
35  -1.05994374      0.73588415
38  -0.77653633     -0.58163769
45  -0.96547460      0.54307608
46  -1.43782030     -1.51354339
```

Step 6: Fitting SVM to the training set

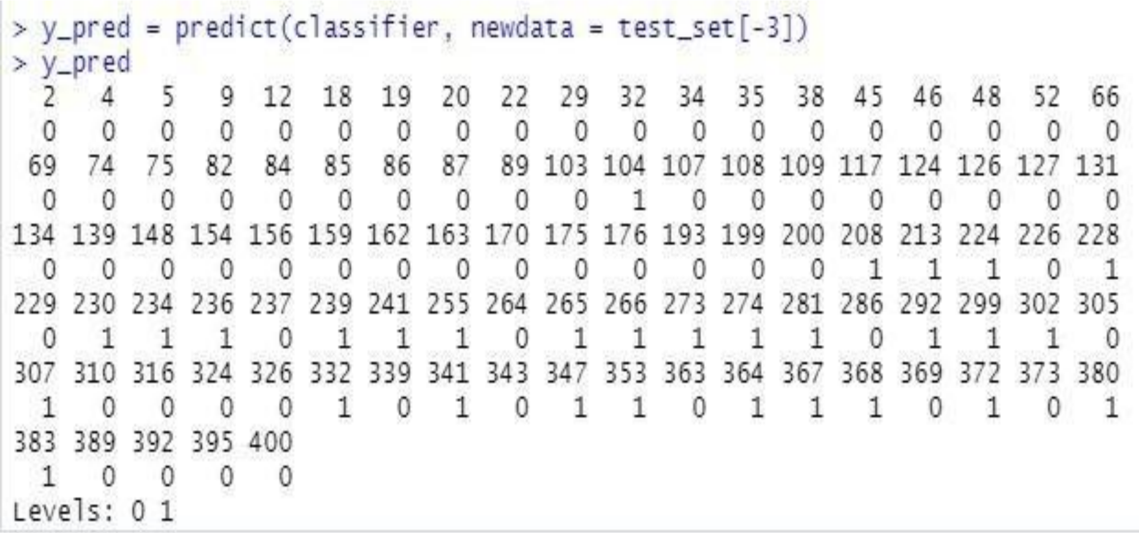


```
400   1.01837732       -0.99938
> install.packages('e1071')
```

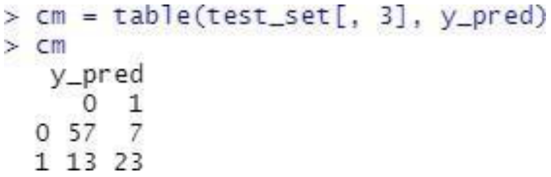


```
> library(e1071)
> classifier = svm(formula = Purchased ~ .,
+                  data = training_set,
+                  type = 'C-classification',
+                  kernel = 'linear')
> classifier

Call:
svm(formula = Purchased ~ ., data = training_set, type = "C-classification",
    kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  116
```

Step 7: Predicting the test set result

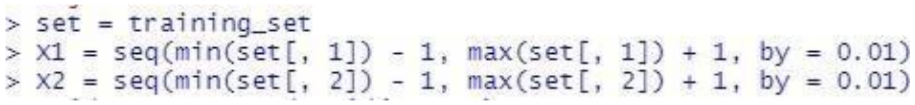


```
> y_pred = predict(classifier, newdata = test_set[-3])
> y_pred
  2   4   5   9  12  18  19  20  22  29  32  34  35  38  45  46  48  52  66 
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 
 69  74  75  82  84  85  86  87  89 103 104 107 108 109 117 124 126 127 131 
  0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0 
134 139 148 154 156 159 162 163 170 175 176 193 199 200 208 213 224 226 228 
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   1   0   1 
229 230 234 236 237 239 241 255 264 265 266 273 274 281 286 292 299 302 305 
  0   1   1   1   0   1   1   1   0   1   1   1   1   1   0   1   1   1   0 
307 310 316 324 326 332 339 341 343 347 353 363 364 367 368 369 372 373 380 
  1   0   0   0   0   1   0   1   0   1   1   0   1   1   1   0   1   0   1 
383 389 392 395 400 
  1   0   0   0   0 
Levels: 0 1
```

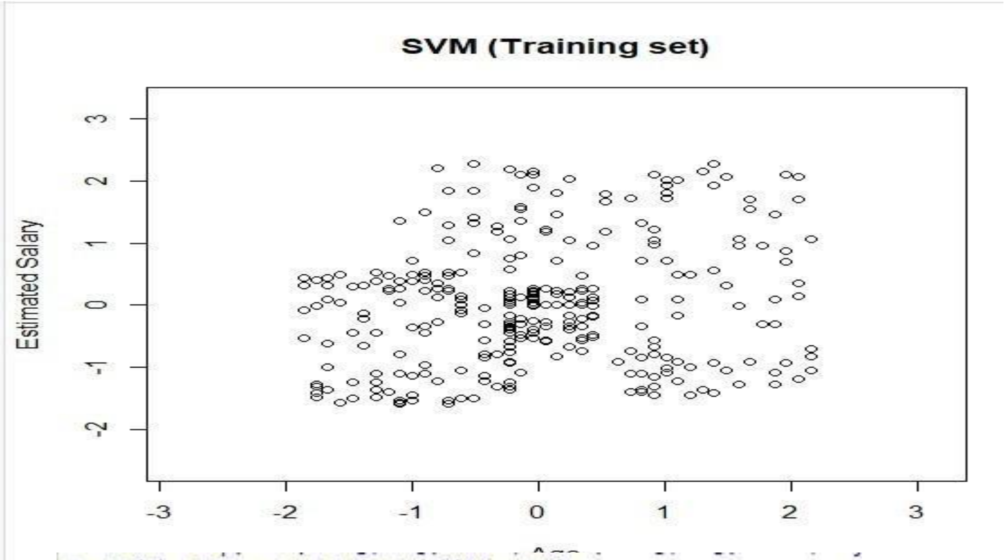


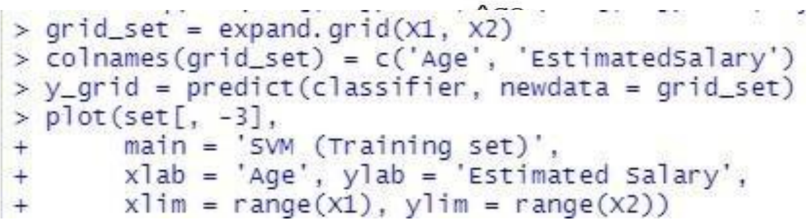
```
> cm = table(test_set[, 3], y_pred)
> cm
   y_pred
     0  1
  0 57  7
  1 13 23
```

Step 8: **Visualizing the Training set results**

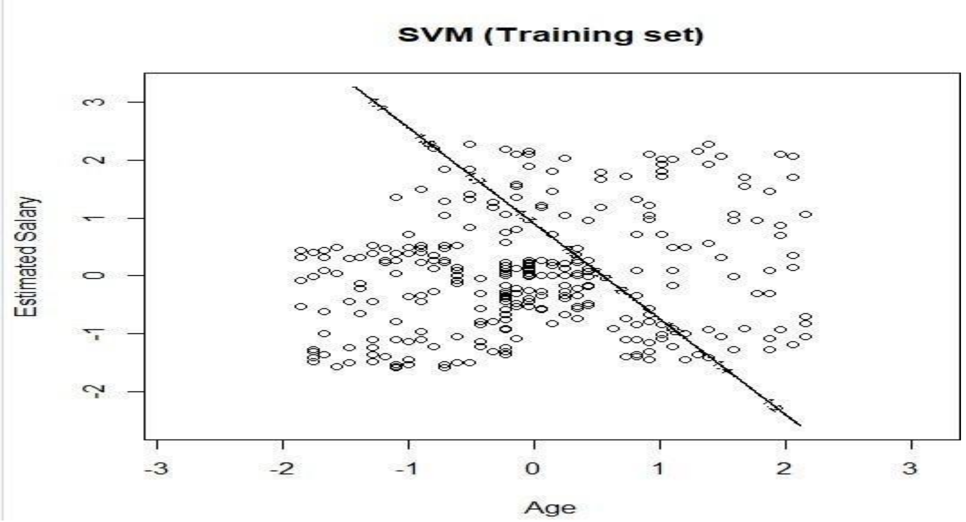


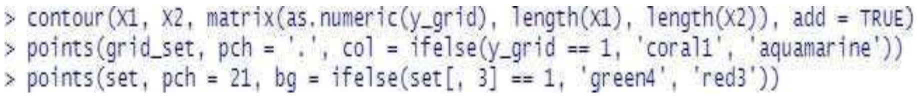
```
> set = training_set
> X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
> X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
```



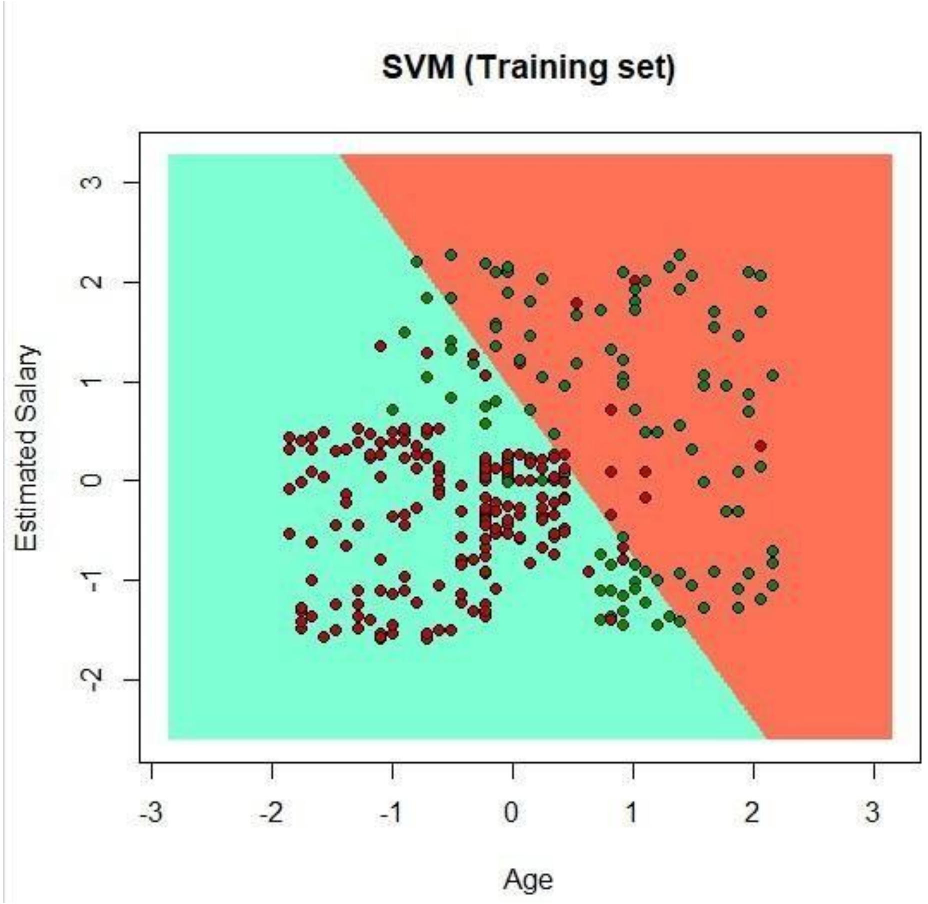


```
> grid_set = expand.grid(X1, X2)
> colnames(grid_set) = c('Age', 'EstimatedSalary')
> y_grid = predict(classifier, newdata = grid_set)
> plot(set[, -3],
+      main = 'SVM (Training set)',
+      xlab = 'Age', ylab = 'Estimated Salary',
+      xlim = range(X1), ylim = range(X2))
```



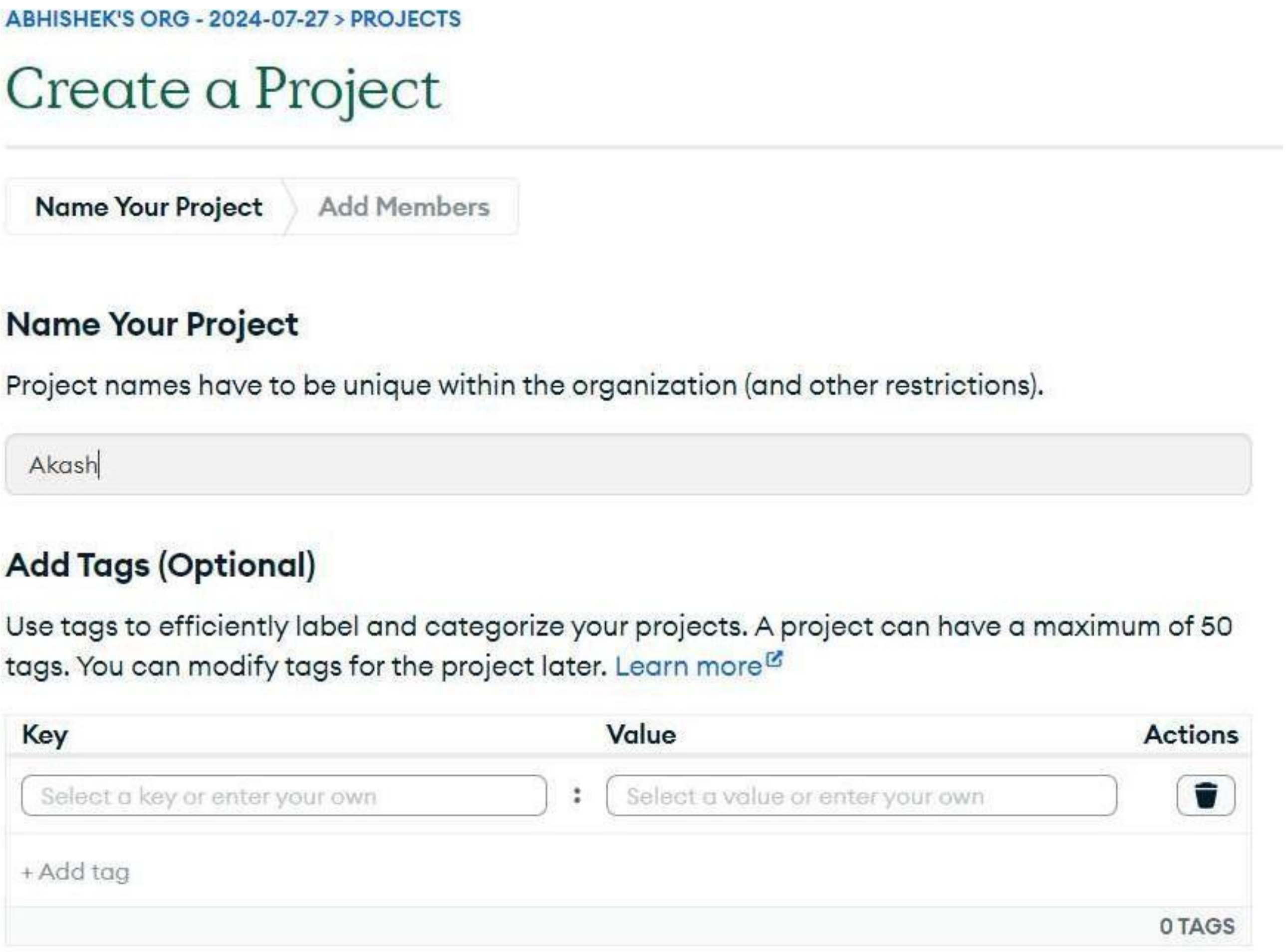


```
> contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
> points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'coral1', 'aquamarine'))
> points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))
```



PRACTICAL NO : 4

Aim: Implement an application that stores big data in Hbase / MongoDB andmanipulate it using R / Python



ABHISHEK'S ORG - 2024-07-27 > PROJECTS

Create a Project

Name Your Project

Project names have to be unique within the organization (and other restrictions).

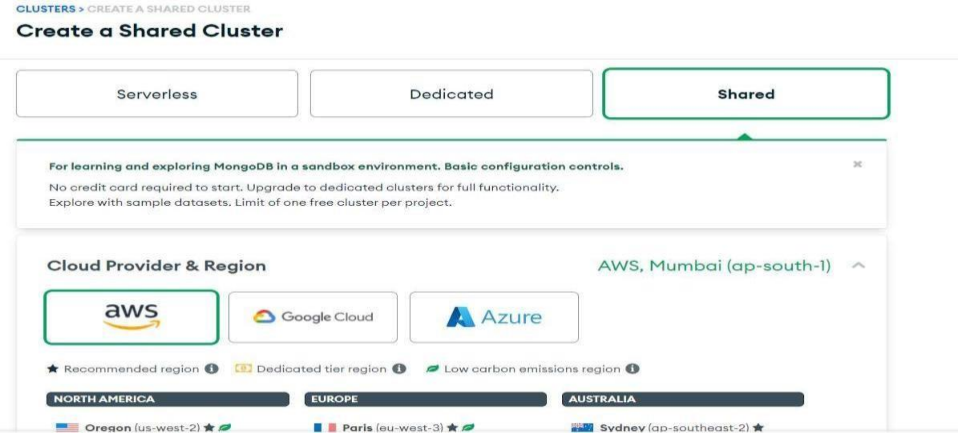
Akash

Add Tags (Optional)

Use tags to efficiently label and categorize your projects. A project can have a maximum of 50 tags. You can modify tags for the project later. Learn more

Key	Value	Actions
Select a key or enter your own	Select a value or enter your own	

Step 1 : Sign up and create a cluster.



CLUSTERS > CREATE A SHARED CLUSTER

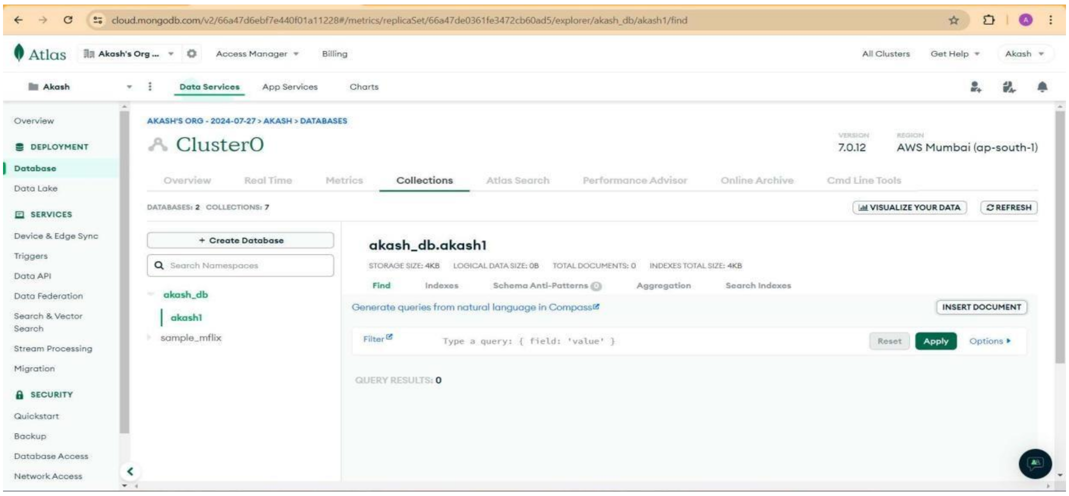
Create a Shared Cluster

For learning and exploring MongoDB in a sandbox environment. Basic configuration controls.

No credit card required to start. Upgrade to dedicated clusters for full functionality.
Explore with sample datasets. Limit of one free cluster per project.

Cloud Provider & Region: AWS, Mumbai (ap-south-1)

This is the home page of mongoDB Atlas.



cloud.mongodb.com/v2/66a47d6ebf7e440f01a11228#/metrics/replicaSet/66a47de0361fe3472cb60ad5/explorer/akash_db/akash1/find

AKASH'S ORG - 2024-07-27 > AKASH > DATABASES

Cluster0

VERSION 7.0.12 REGION AWS Mumbai (ap-south-1)

DATABASES: 2 COLLECTIONS: 7

akash_db.akash1

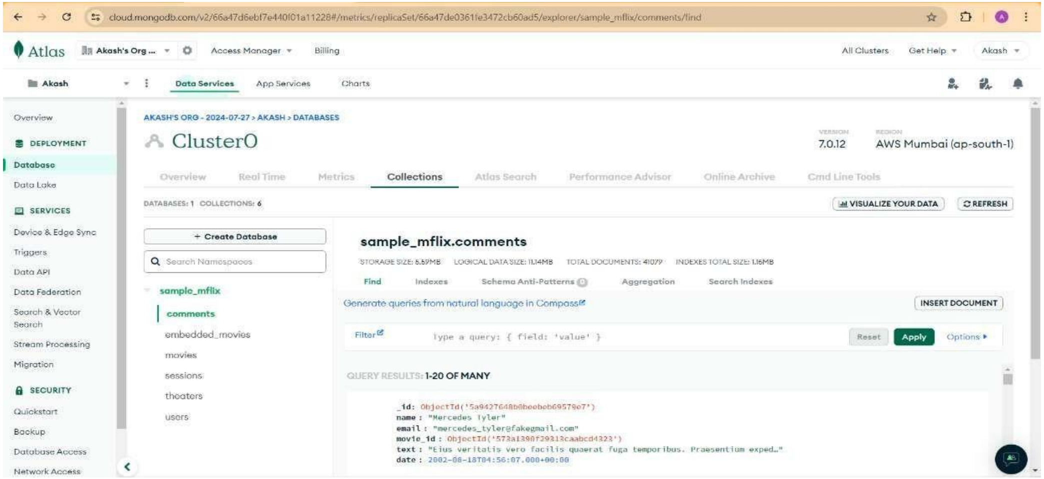
STORAGE SIZE: 4KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' }

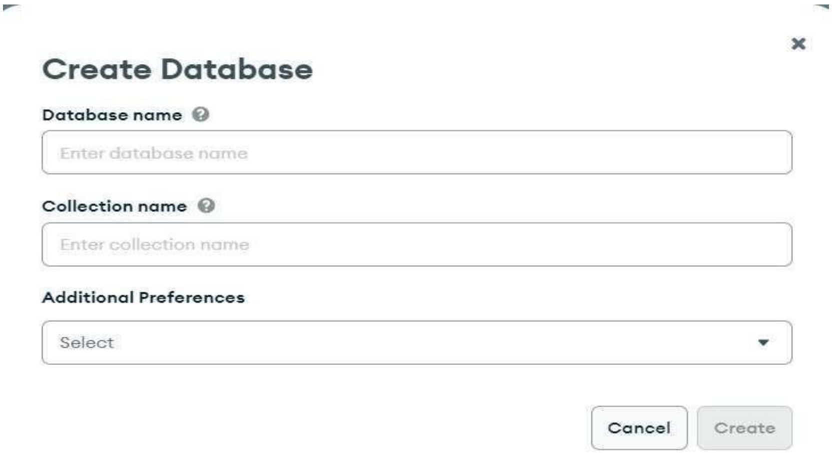
QUERY RESULTS: 0

Step 2 : Click on collections to create and view existing databases.

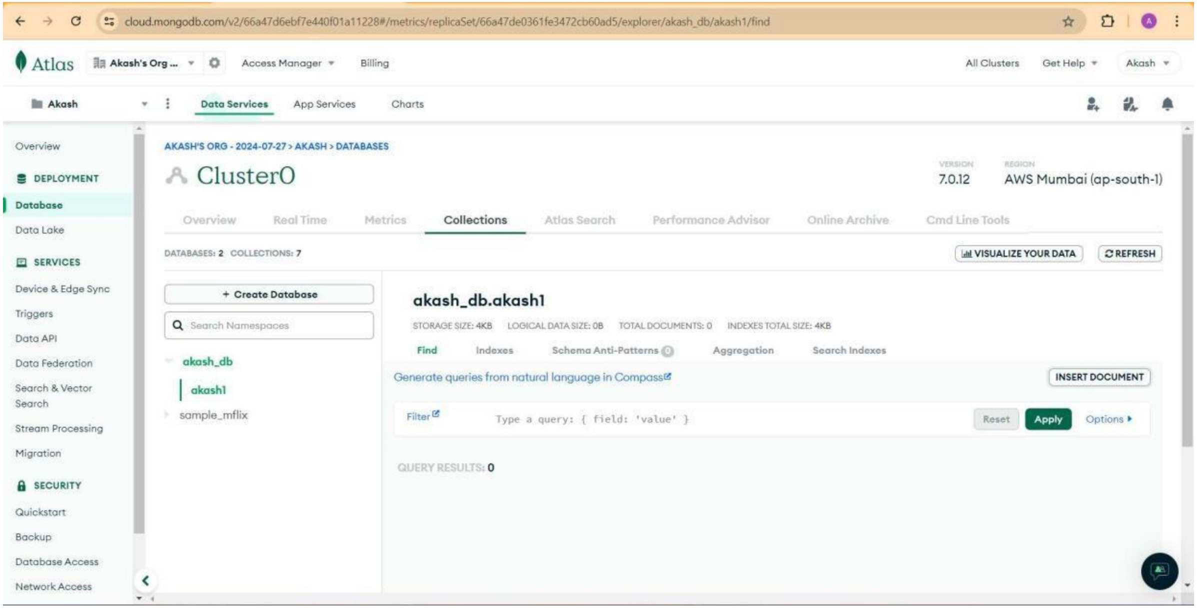


```
_id: ObjectId('5a9427648b0beebeb69579e7')
name : "Mercedes Tyler"
email : "mercedes_tyler@fakegmail.com"
movie_id : ObjectId('573a1390f29313caabcd4323')
text : "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped…"
date : 2002-08-18T04:56:07.000+00:00
```

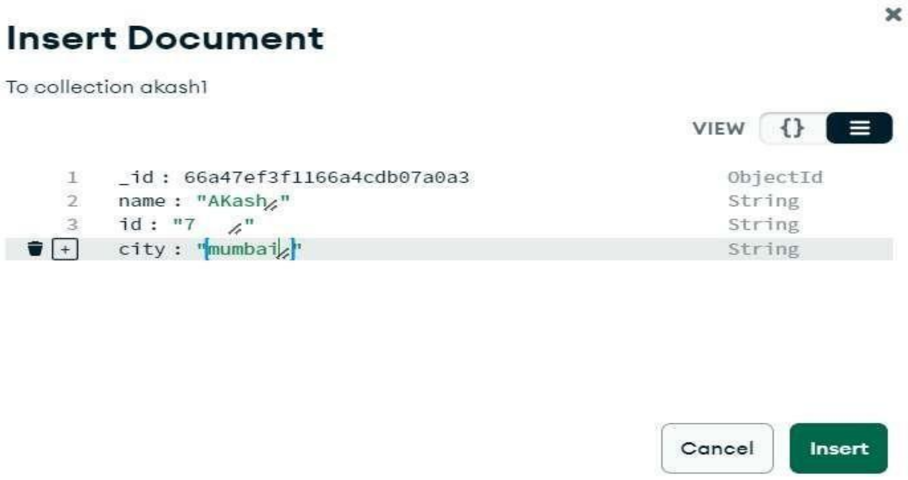
Step 3 : Click on 'Add My Own Data' to create a database.



Step 4 : Click on insert document to add records.



Since MongoDB is a No-SQL database, so you can add ‘n’ number of columns for any row/record.

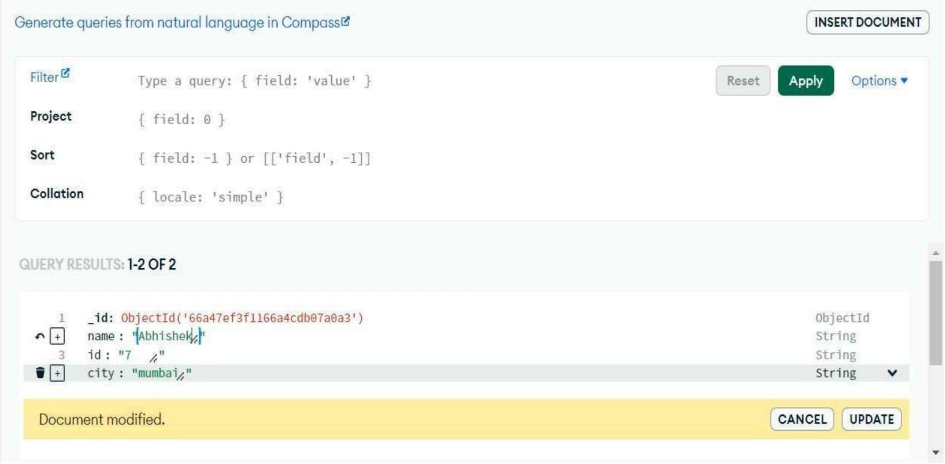


Insert Document

To collection akash1

```
_id : 66a47ef3f1166a4cdb07a0a3        ObjectId
name : "AKash"                        String
id : "7"                              String
city : "mumbai"                       String
```

Perform updating data



Generate queries from natural language in Compass

INSERT DOCUMENT

Filter: Type a query: { field: 'value' }

Project: { field: 0 }

Sort: { field: -1 } or [['field', -1]]

Collation: { locale: 'simple' }

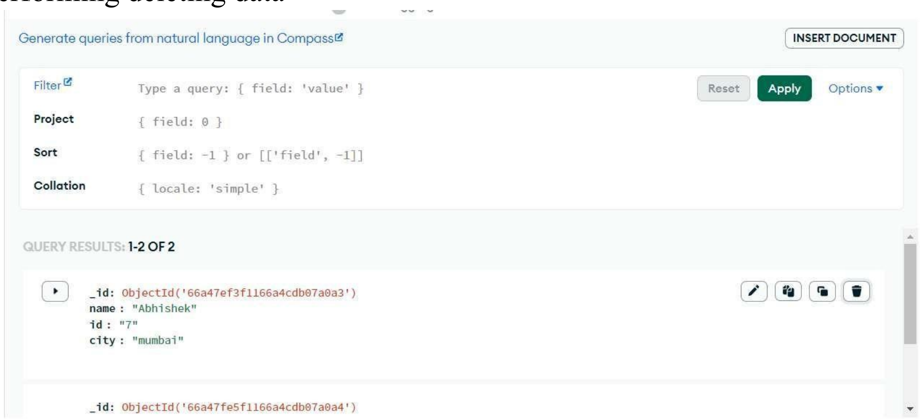
Reset | Apply | Options

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('66a47ef3f1166a4cdb07a0a3')    ObjectId
name : "Abhishek"                             String
id : "7"                                      String
city : "mumbai"                               String
```

Document modified. CANCEL UPDATE

Performing deleting data



Generate queries from natural language in Compass

INSERT DOCUMENT

Filter: Type a query: { field: 'value' }

Project: { field: 0 }

Sort: { field: -1 } or [['field', -1]]

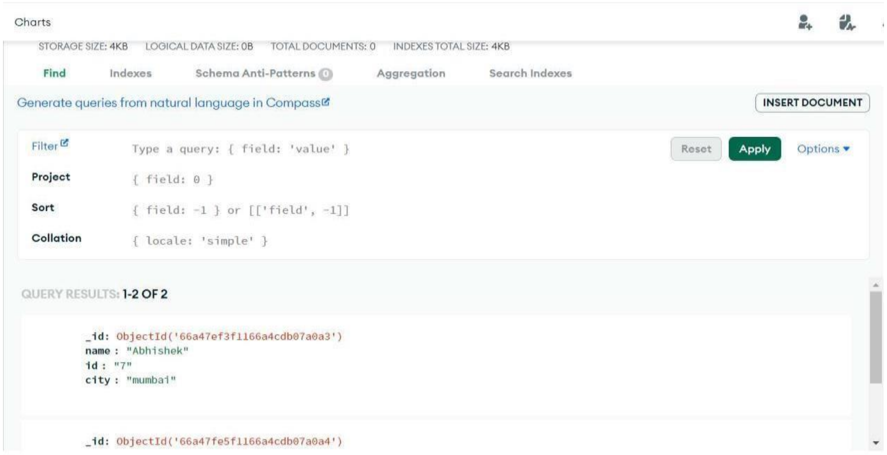
Collation: { locale: 'simple' }

Reset | Apply | Options

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('66a47ef3f1166a4cdb07a0a3')
name : "Abhishek"
id : "7"
city : "mumbai"

_id: ObjectId('66a47fe5f1166a4cdb07a0a4')
```

Performing Insert data



Charts

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB

Find | Indexes | Schema Anti-Patterns 0 | Aggregation | Search Indexes

Generate queries from natural language in Compass

INSERT DOCUMENT

Filter: Type a query: { field: 'value' }

Project: { field: 0 }

Sort: { field: -1 } or [['field', -1]]

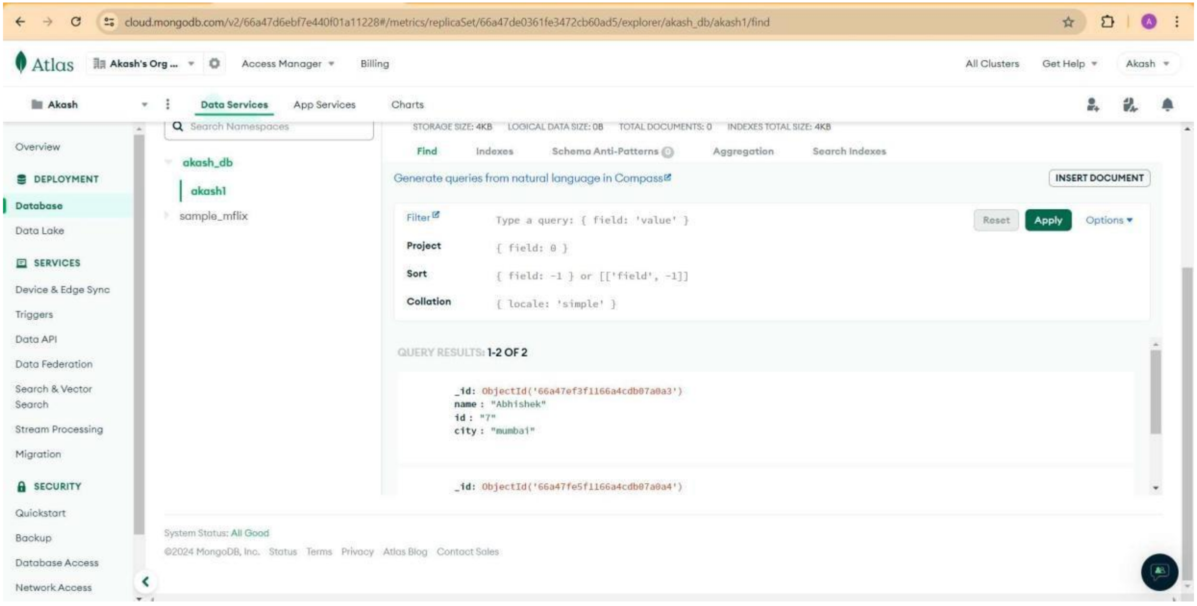
Collation: { locale: 'simple' }

Reset | Apply | Options

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('66a47ef3f1166a4cdb07a0a3')
name : "Abhishek"
id : "7"
city : "mumbai"

_id: ObjectId('66a47fe5f1166a4cdb07a0a4')
```

Step 5 : To start with the connection click on Overview, and then click on Connect.



Filter `Type a query: { field: 'value' }`

Project `{ field: 0 }`

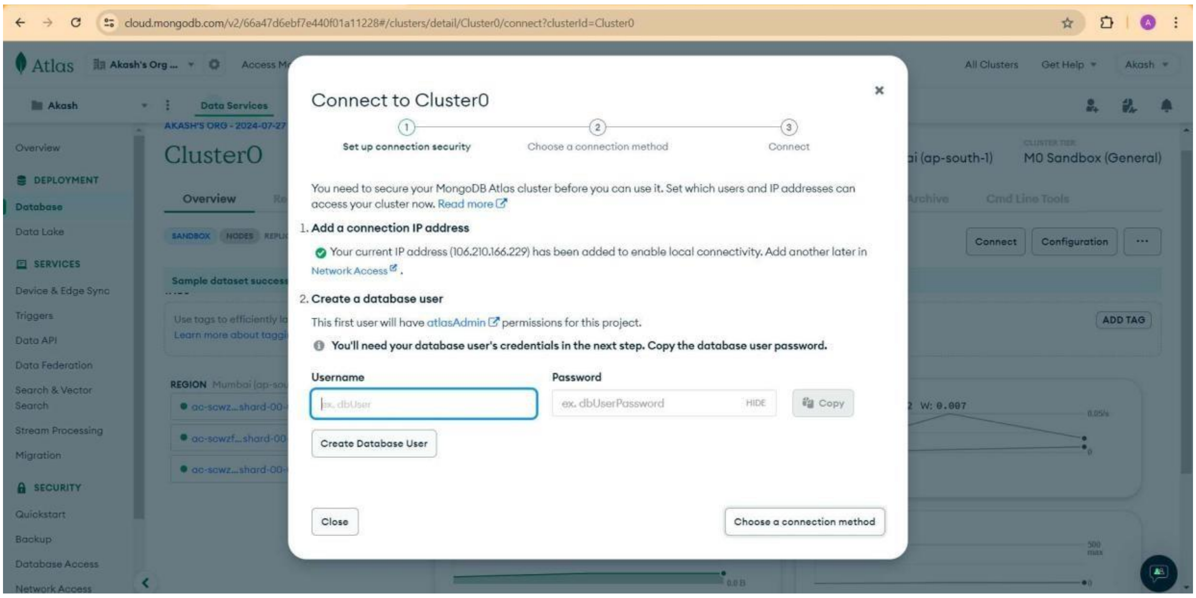
Sort `{ field: -1 } or [['field', -1]]`

Collation `{ locale: 'simple' }`

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('66a47ef3f1166a4cdb07a0a3')
name : "Abhishek"
id : "7"
city : "mumbai"

_id: ObjectId('66a47fe5f1166a4cdb07a0a4')
```

Step 6 : Select on add your current IP and create a MongoDB user.



Connect to Cluster0

① Set up connection security — ② Choose a connection method — ③ Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. Read more

1. Add a connection IP address

✓ Your current IP address (106.210.166.229) has been added to enable local connectivity. Add another later in Network Access .

2. Create a database user

This first user will have atlasAdmin permissions for this project.

You'll need your database user's credentials in the next step. Copy the database user password.

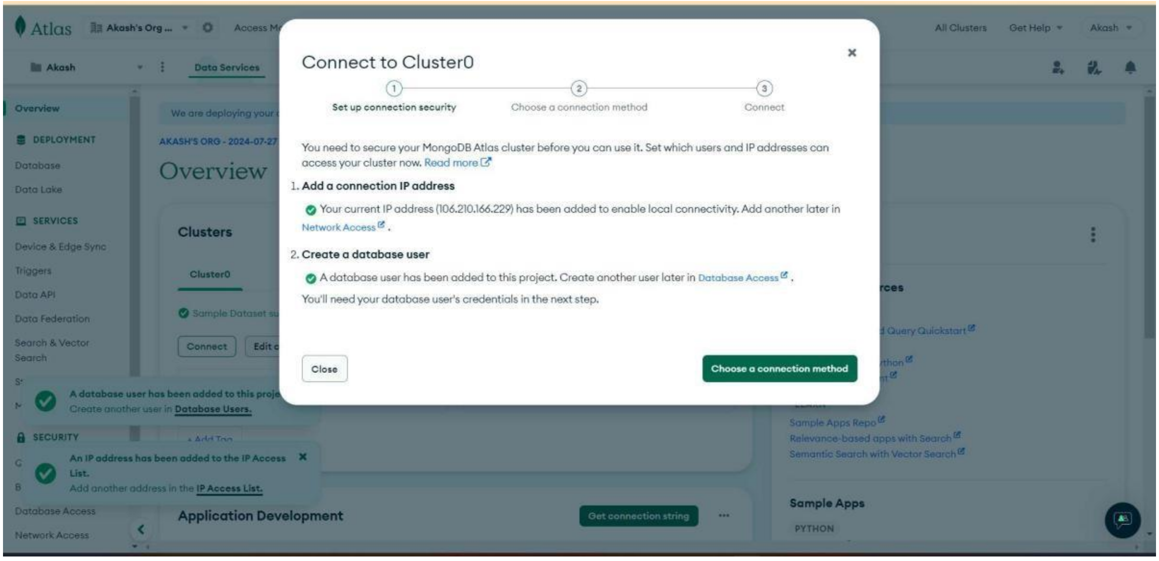
Username: ex. dbUser

Password: ex. dbUserPassword HIDE Copy

Create Database User

Close | Choose a connection method

Step 7 : Click on ‘Connect your application’.



A database user has been added to this project. Create another user in Database Users.

An IP address has been added to the IP Access List. Add another address in the IP Access List.

Connect to Cluster0

① Set up connection security — ② Choose a connection method — ③ Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. Read more

1. Add a connection IP address

✓ Your current IP address (106.210.166.229) has been added to enable local connectivity. Add another later in Network Access .

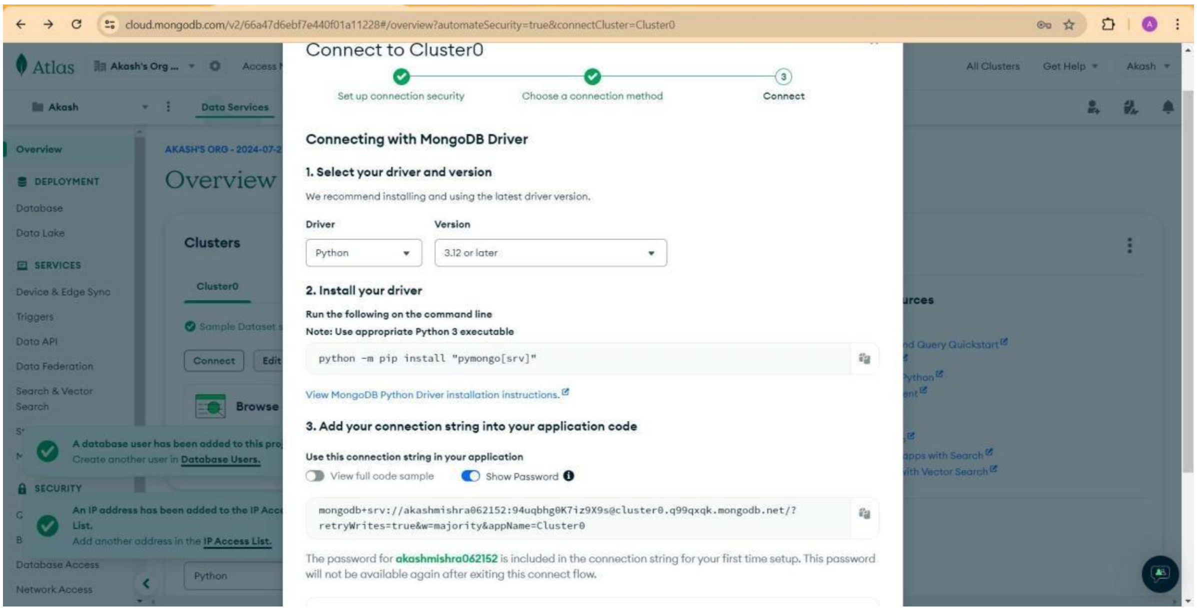
2. Create a database user

✓ A database user has been added to this project. Create another user later in Database Access .

You'll need your database user's credentials in the next step.

Close | Choose a connection method

Step 8 : Select the driver as ‘Python’ and version as ‘3.6 or later’. (Select the version as 3.6 or later only if your Python’s version is 3.6 or later.)



Connect to Cluster0

Connecting with MongoDB Driver

1. Select your driver and version

We recommend installing and using the latest driver version.

Driver: Python

Version: 3.12 or later

2. Install your driver

Run the following on the command line

Note: Use appropriate Python 3 executable

```
python -m pip install "pymongo[srv]"
```

View MongoDB Python Driver installation instructions.

3. Add your connection string into your application code

Use this connection string in your application

○ View full code sample ● Show Password

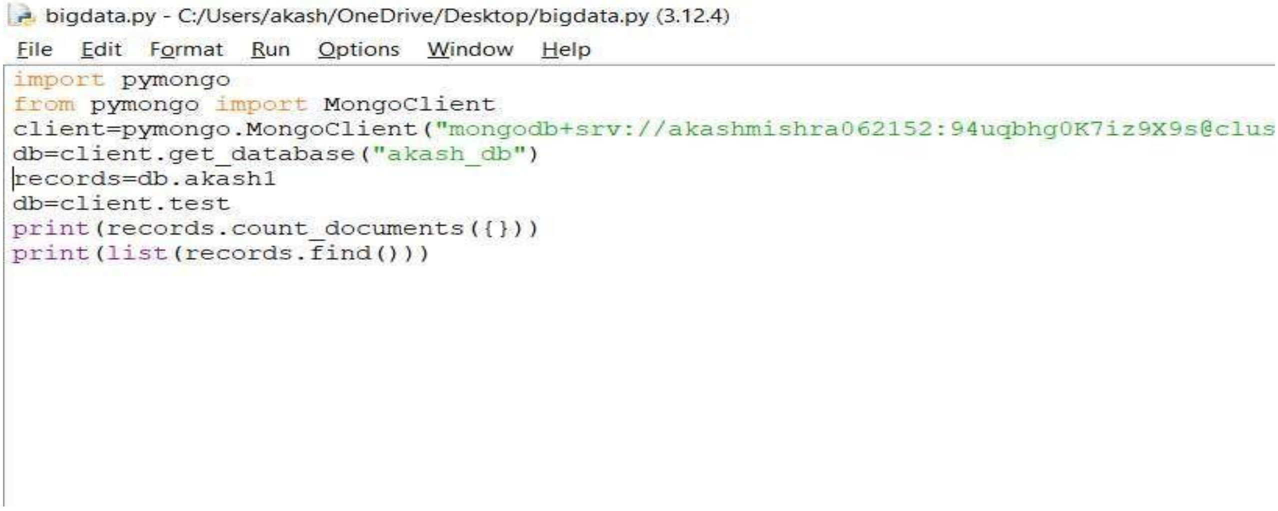
```
mongodb+srv://akashmishra062152:94uqbhg0K7iz9X9s@cluster0.q99qxqk.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
```

The password for akashmishra062152 is included in the connection string for your first time setup. This password will not be available again after exiting this connect flow.

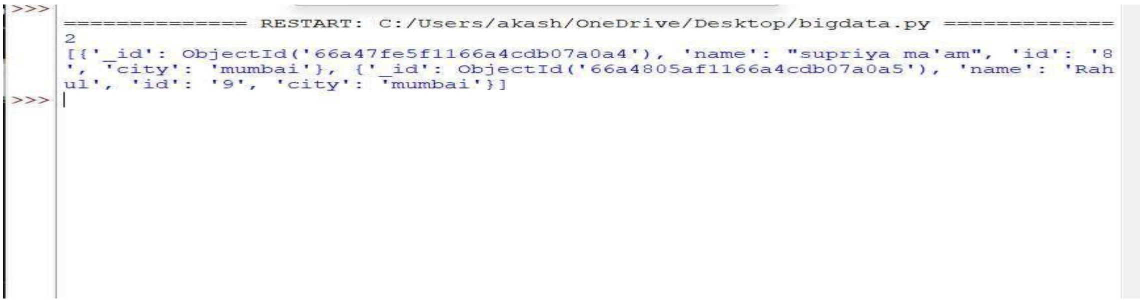
A database user has been added to this proj Create another user in Database Users.

An IP address has been added to the IP Acce List. Add another address in the IP Access List.

Step 9 : Write the code given below in a Python file.


bigdata.py - C:/Users/akash/OneDrive/Desktop/bigdata.py (3.12.4)
File Edit Format Run Options Window Help

```
import pymongo
from pymongo import MongoClient
client=pymongo.MongoClient("mongodb+srv://akashmishra062152:94uqbhg0K7iz9X9s@clus
db=client.get_database("akash_db")
records=db.akash1
db=client.test
print(records.count_documents({}))
print(list(records.find()))
```



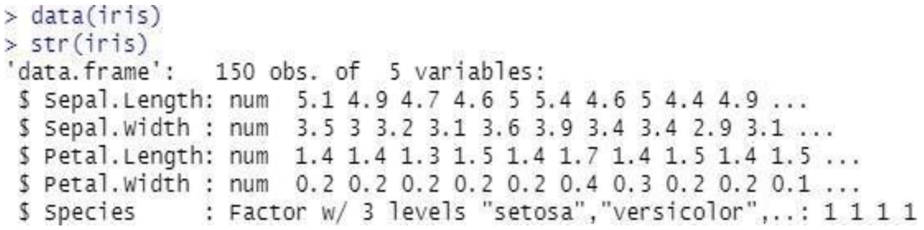
```
>>>
=============== RESTART: C:/Users/akash/OneDrive/Desktop/bigdata.py ==============
2
[{'_id': ObjectId('66a47fe5f1166a4cdb07a0a4'), 'name': "supriya ma'am", 'id': '8
', 'city': 'mumbai'}, {'_id': ObjectId('66a4805af1166a4cdb07a0a5'), 'name': 'Rah
ul', 'id': '9', 'city': 'mumbai'}]
>>>
```

Output :

PRACTICAL NO : 5

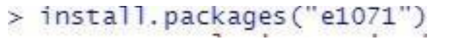
Aim: **write program in R of Naive baye's theorem**

# Loading data

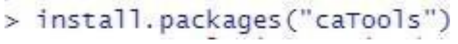


```
> data(iris)
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1
```

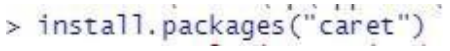
# Installing Packages



```
> install.packages("e1071")
```

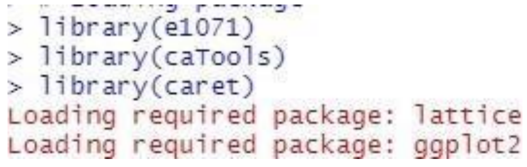


```
> install.packages("caTools")
```



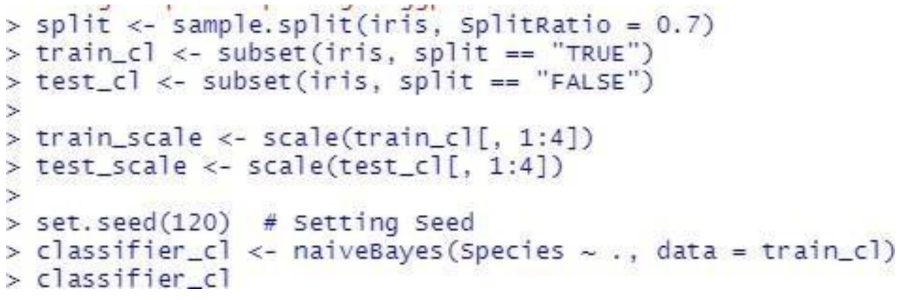
```
> install.packages("caret")
```

# Loading package

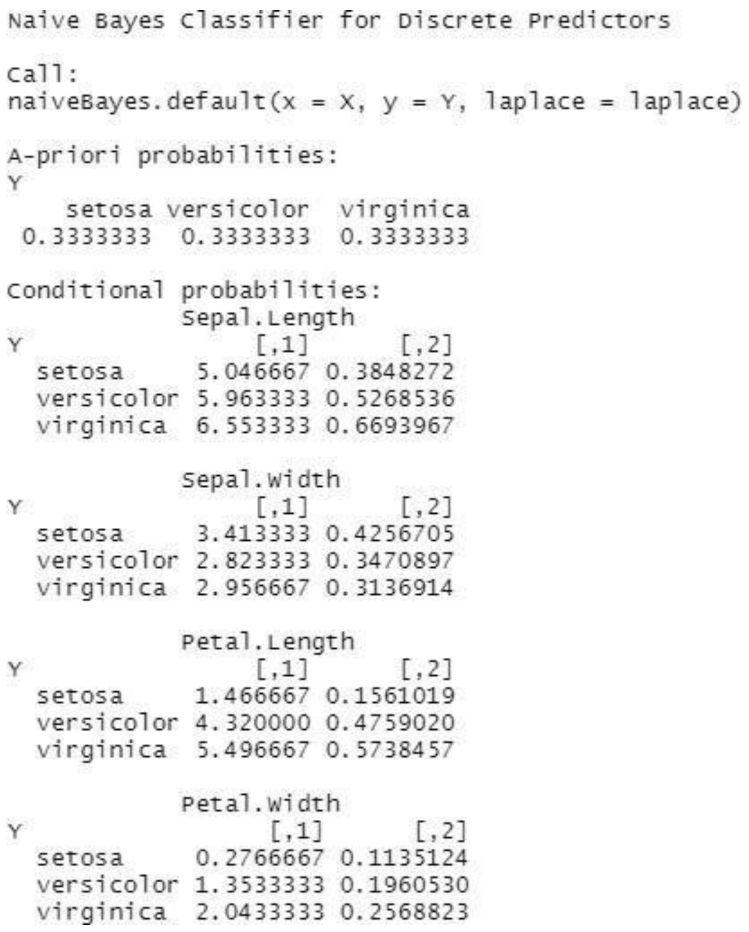


```
> library(e1071)
> library(caTools)
> library(caret)
Loading required package: lattice
Loading required package: ggplot2
```

# Splitting data into train and test data

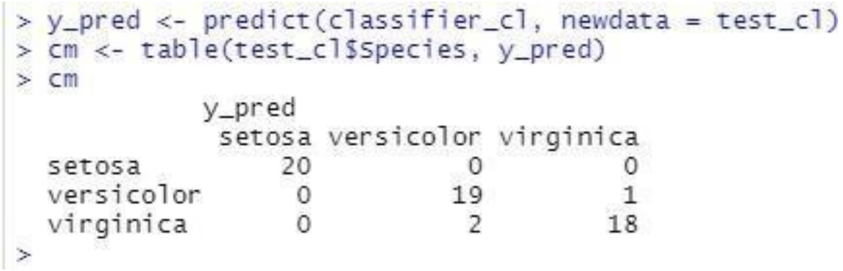


```
> split <- sample.split(iris, SplitRatio = 0.7)
> train_cl <- subset(iris, split == "TRUE")
> test_cl <- subset(iris, split == "FALSE")
>
> train_scale <- scale(train_cl[, 1:4])
> test_scale <- scale(test_cl[, 1:4])
>
> set.seed(120)  # Setting Seed
> classifier_cl <- naiveBayes(Species ~ ., data = train_cl)
> classifier_cl
```



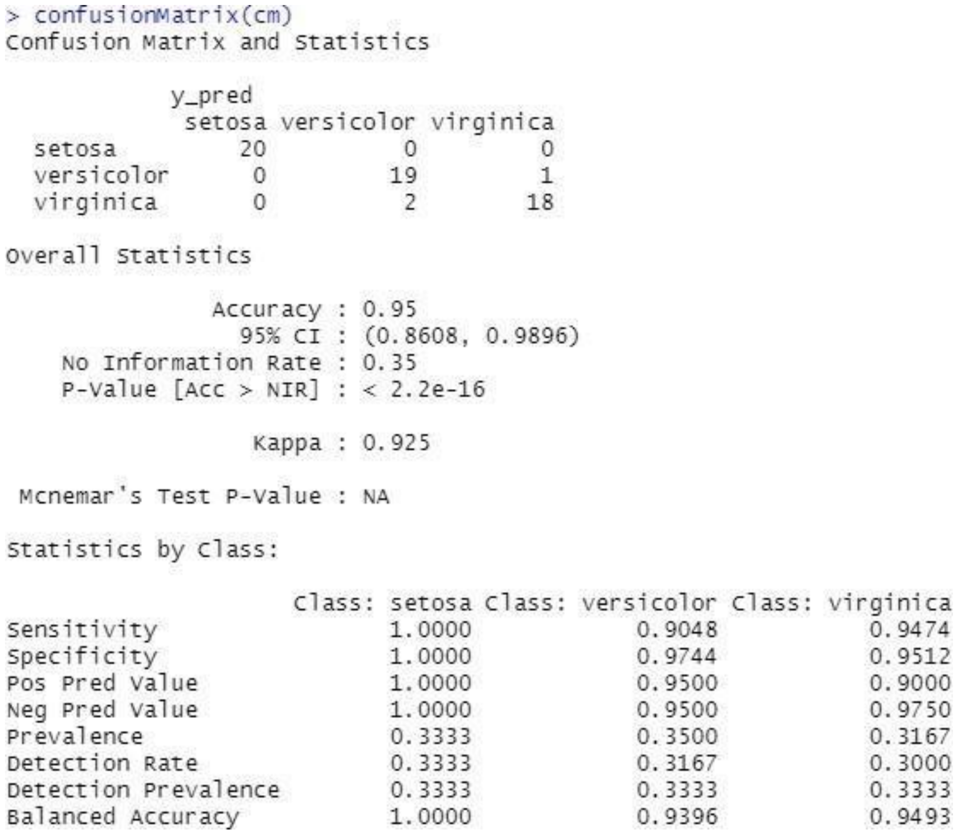
```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    setosa versicolor  virginica 
 0.3333333  0.3333333  0.3333333 

Conditional probabilities:
            Sepal.Length
Y                [,1]      [,2]
  setosa     5.046667 0.3848272
  versicolor 5.963333 0.5268536
  virginica  6.553333 0.6693967

            Sepal.Width
Y                [,1]      [,2]
  setosa     3.413333 0.4256705
  versicolor 2.823333 0.3470897
  virginica  2.956667 0.3136914

            Petal.Length
Y                [,1]      [,2]
  setosa     1.466667 0.1561019
  versicolor 4.320000 0.4759020
  virginica  5.496667 0.5738457

            Petal.Width
Y                 [,1]      [,2]
  setosa     0.2766667 0.1135124
  versicolor 1.3533333 0.1960530
  virginica  2.0433333 0.2568823
```

# Predicting on test data'



```
> y_pred <- predict(classifier_cl, newdata = test_cl)
> cm <- table(test_cl$Species, y_pred)
> cm
            y_pred
             setosa versicolor virginica
  setosa         20          0         0
  versicolor      0         19         1
  virginica       0          2        18
> 
```

# Model Evauation



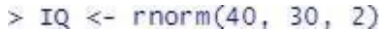
```
> confusionMatrix(cm)
Confusion Matrix and Statistics

            y_pred
             setosa versicolor virginica
  setosa         20          0         0
  versicolor      0         19         1
  virginica       0          2        18

Overall Statistics

               Accuracy : 0.95
                 95% CI : (0.8608, 0.9896)
    No Information Rate : 0.35
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.925

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9048           0.9474
Specificity                 1.0000            0.9744           0.9512
Pos Pred Value              1.0000            0.9500           0.9000
Neg Pred Value              1.0000            0.9500           0.9750
Prevalence                  0.3333            0.3500           0.3167
Detection Rate              0.3333            0.3167           0.3000
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            0.9396           0.9493
```

PRACTICAL NO : 6

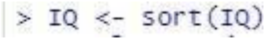
Aim: WAP showing implementation of Regression model.

Regression is a method to mathematically formulate relationship between variables thatin due course can be used to estimate, interpolate and extrapolate. Suppose we want to estimate the weight of individuals, which is influenced by height, diet, workout, etc.
Here, *Weight* is the **predicted** variable

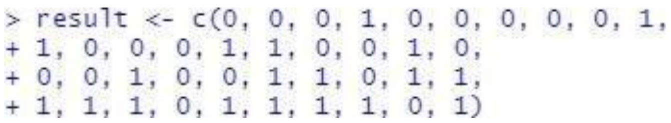
Lets implementation of Regression Model some Example:



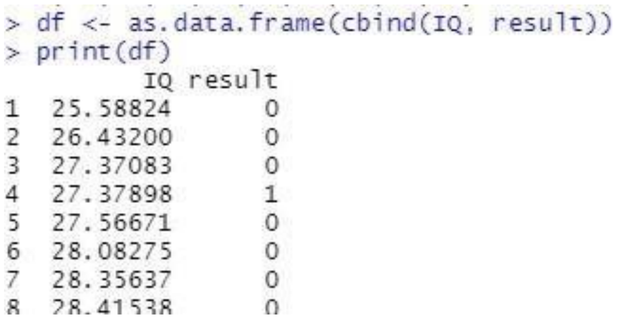
```
> IQ <- rnorm(40, 30, 2)
```



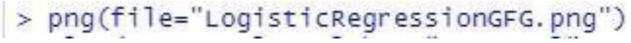
```
> IQ <- sort(IQ)
```



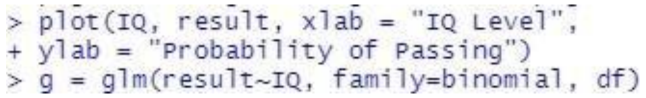
```
> result <- c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
+ 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
+ 0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
+ 1, 1, 1, 0, 1, 1, 1, 1, 0, 1)
```



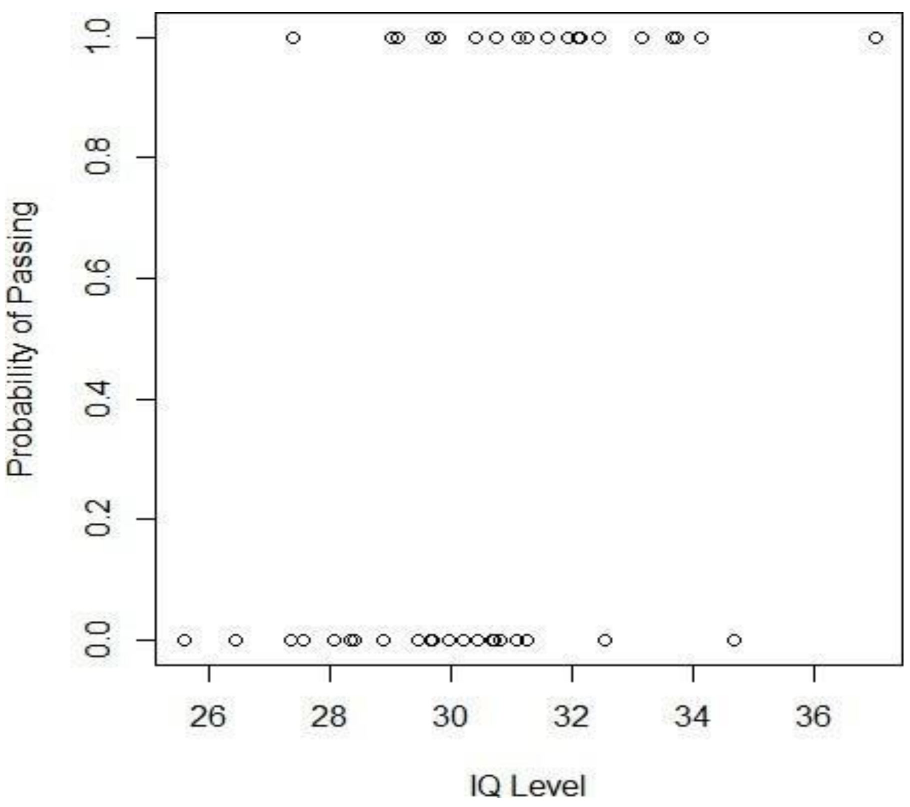
```
> df <- as.data.frame(cbind(IQ, result))
> print(df)
        IQ result
1  25.58824      0
2  26.43200      0
3  27.37083      0
4  27.37898      1
5  27.56671      0
6  28.08275      0
7  28.35637      0
8  28.41538      0
```

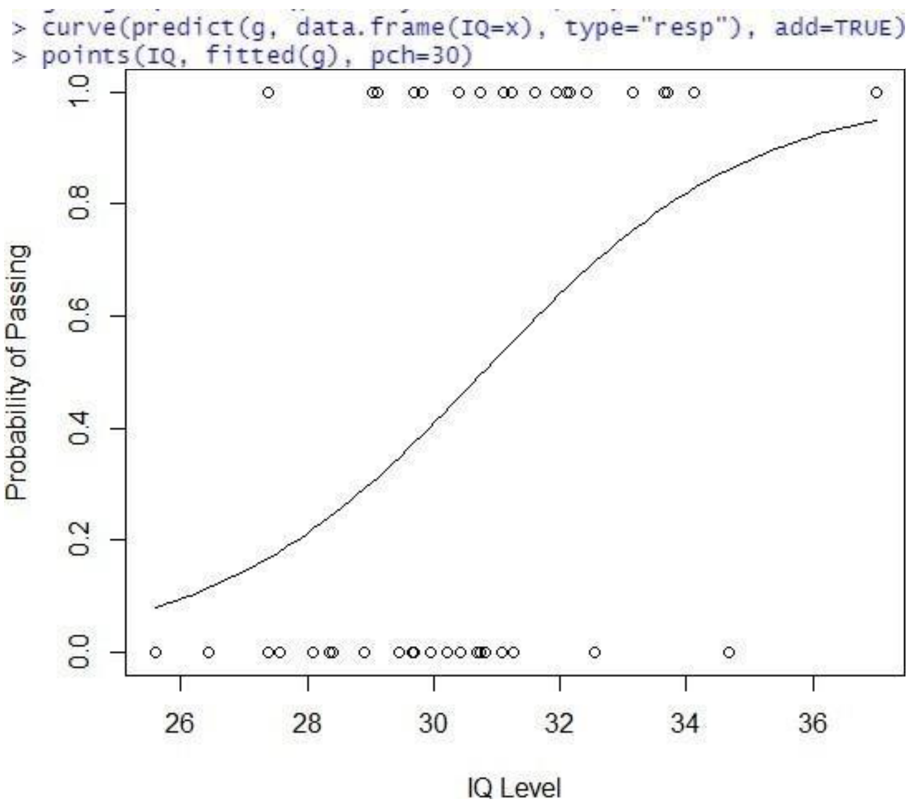


```
> png(file="LogisticRegressionGFG.png")
```

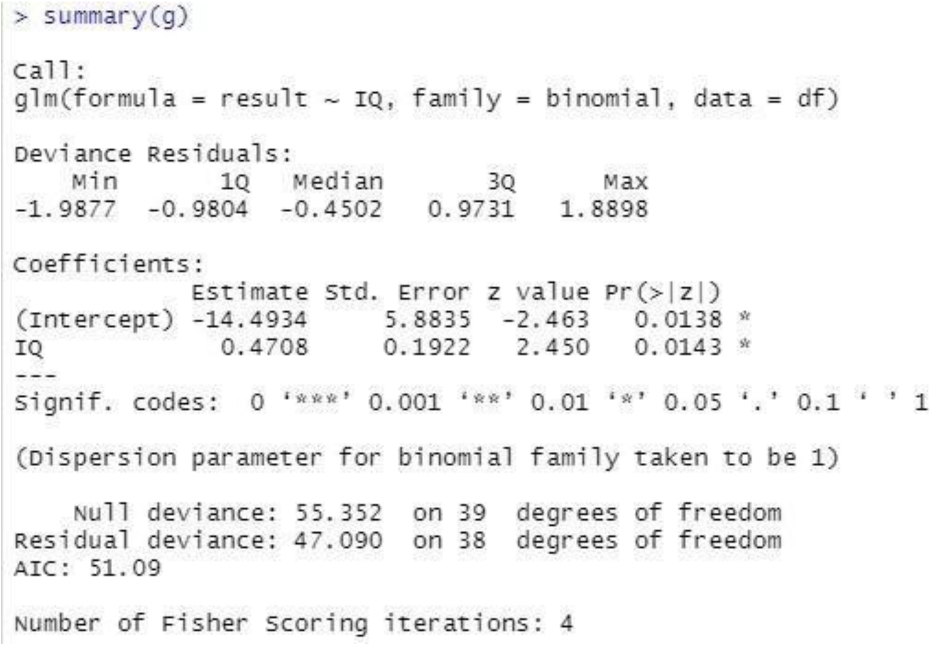


```
> plot(IQ, result, xlab = "IQ Level",
+ ylab = "Probability of Passing")
> g = glm(result~IQ, family=binomial, df)
```

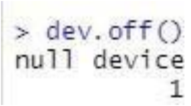




```
> curve(predict(g, data.frame(IQ=x), type="resp"), add=TRUE)
> points(IQ, fitted(g), pch=30)
```



```
> summary(g)

Call:
glm(formula = result ~ IQ, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9877  -0.9804  -0.4502   0.9731   1.8898

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.4934     5.8835  -2.463   0.0138 *
IQ            0.4708     0.1922   2.450   0.0143 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 55.352  on 39  degrees of freedom
Residual deviance: 47.090  on 38  degrees of freedom
AIC: 51.09

Number of Fisher Scoring iterations: 4
```

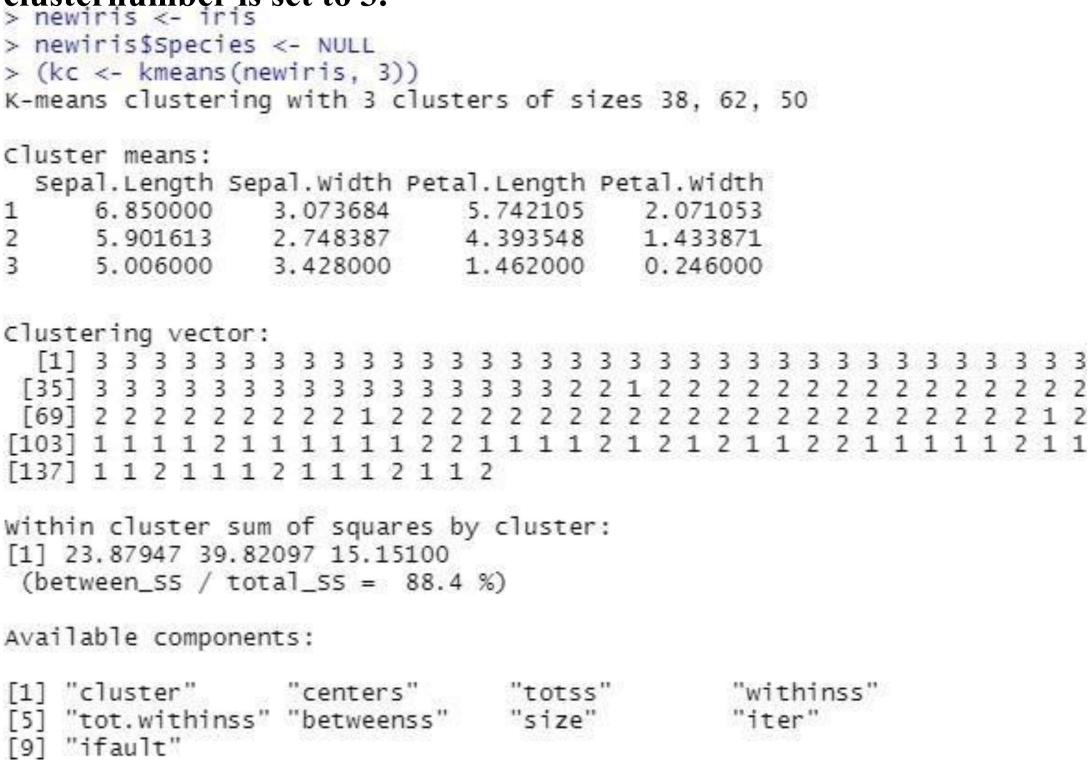


```
> dev.off()
null device
          1
```

PRACTICAL NO : 7

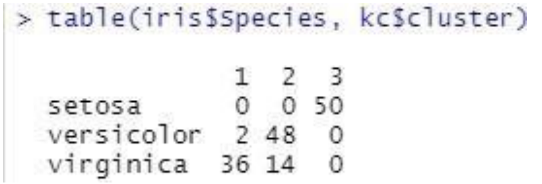
Aim: WAP showing clustering.

Step 1: Apply kmeans to *newiris*, and store the clustering result in *kc*. The clusternumber is set to 3.



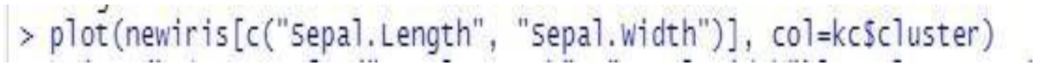
```
> newiris <- iris
> newiris$Species <- NULL
> (kc <- kmeans(newiris, 3))
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [35] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [69] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
[103] 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1
[137] 1 1 2 1 1 1 2 1 1 1 2 1 1 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

Step 2: Compare the Species label with the clustering result

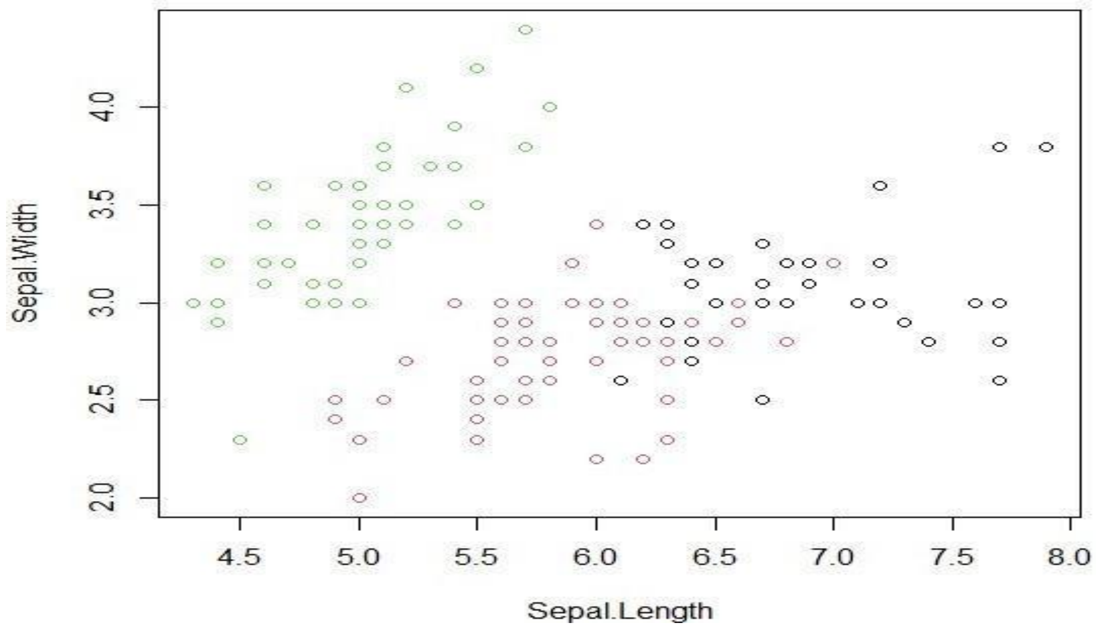


```
> table(iris$Species, kc$cluster)

              1  2  3
  setosa      0  0 50
  versicolor  2 48  0
  virginica  36 14  0
```

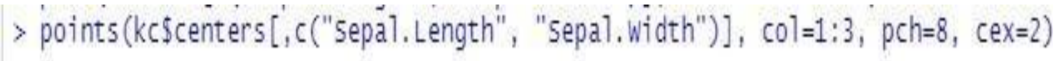
Step 3 : Plot the clusters and their centres. Note that there are four dimensions in thedata and that only the first two dimensions are used to draw the plot below.



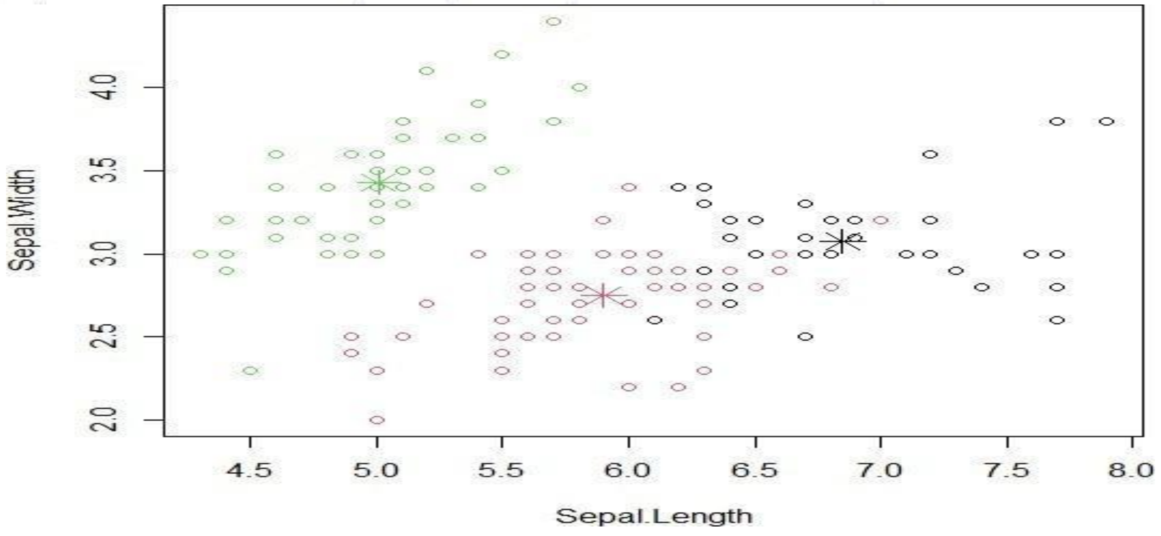
```
> plot(newiris[c("Sepal.Length", "Sepal.Width")], col=kc$cluster)
```



Step 4: Some black points close to the green centre (asterisk) are actually closer to theblack centre in the four dimensional space.



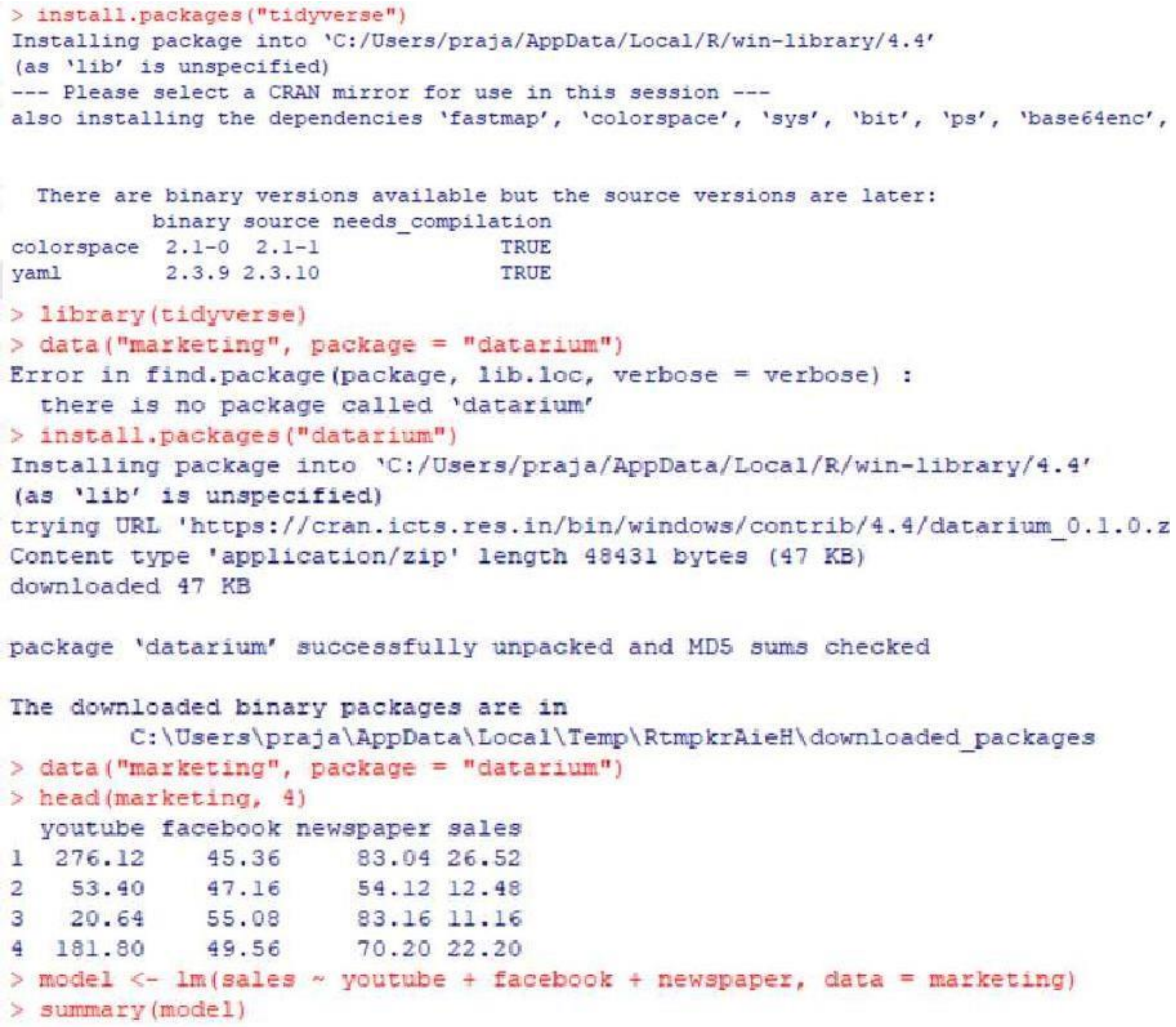
```
> points(kc$centers[,c("Sepal.Length", "Sepal.Width")], col=1:3, pch=8, cex=2)
```



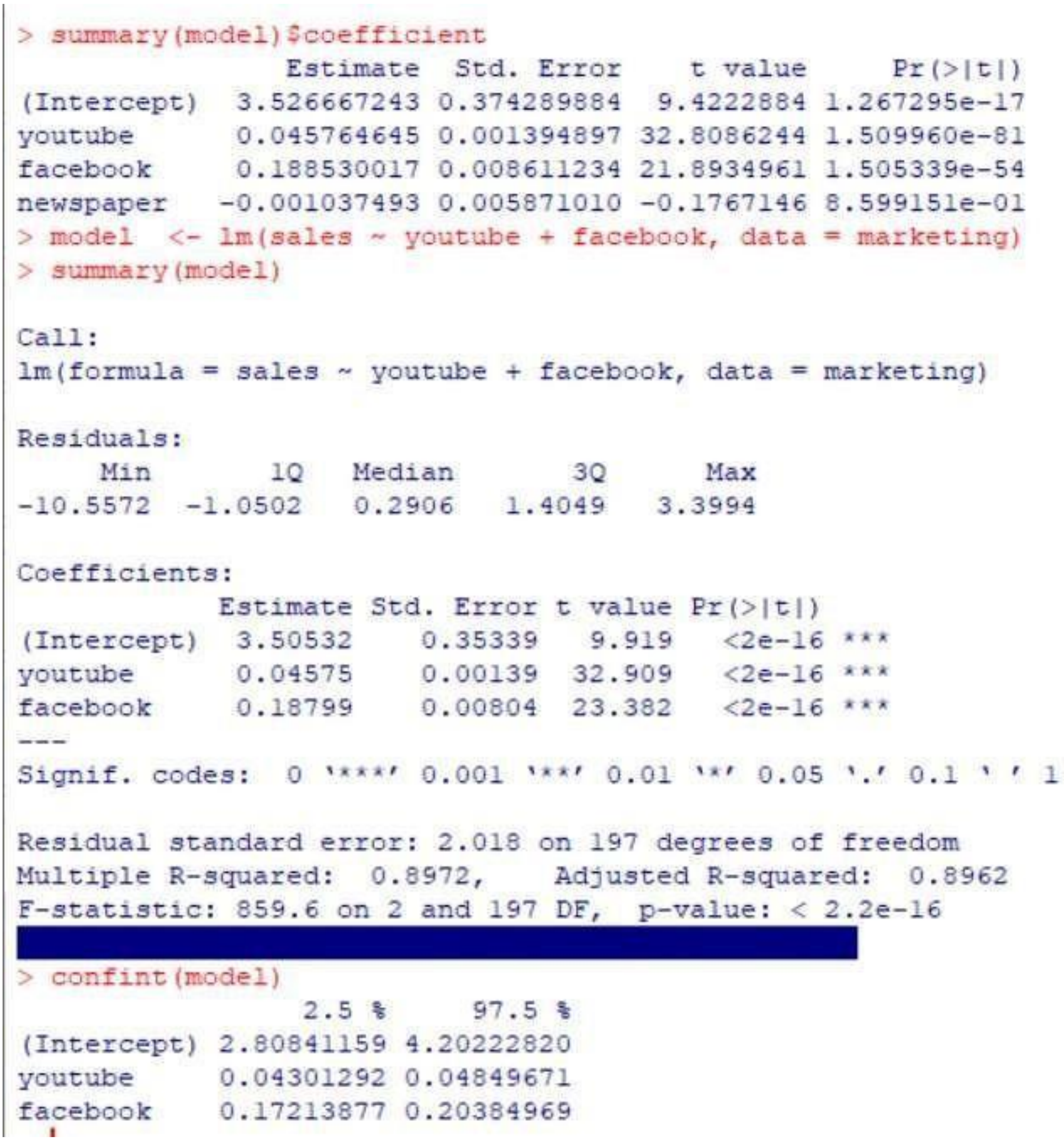
PRACTICAL NO : 8

Multiple regression

Aim: Apply Multiple regressions, if data have a continuous independent variable .



```
> install.packages("tidyverse")
Installing package into 'C:/Users/praja/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
also installing the dependencies 'fastmap', 'colorspace', 'sys', 'bit', 'ps', 'base64enc',


  There are binary versions available but the source versions are later:
           binary source needs_compilation
colorspace  2.1-0  2.1-1              TRUE
yaml        2.3.9 2.3.10              TRUE

> library(tidyverse)
> data("marketing", package = "datarium")
Error in find.package(package, lib.loc, verbose = verbose) :
  there is no package called 'datarium'
> install.packages("datarium")
Installing package into 'C:/Users/praja/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
trying URL 'https://cran.icts.res.in/bin/windows/contrib/4.4/datarium_0.1.0.z
Content type 'application/zip' length 48431 bytes (47 KB)
downloaded 47 KB

package 'datarium' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\praja\AppData\Local\Temp\RtmpkrAieH\downloaded_packages
> data("marketing", package = "datarium")
> head(marketing, 4)
  youtube facebook newspaper sales
1  276.12    45.36     83.04 26.52
2   53.40    47.16     54.12 12.48
3   20.64    55.08     83.16 11.16
4  181.80    49.56     70.20 22.20
> model <- lm(sales ~ youtube + facebook + newspaper, data = marketing)
> summary(model)
```



```
> summary(model)$coefficient
                Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)  3.526667243 0.374289884  9.4222884 1.267295e-17
youtube      0.045764645 0.001394897 32.8086244 1.509960e-81
facebook     0.188530017 0.008611234 21.8934961 1.505339e-54
newspaper   -0.001037493 0.005871010 -0.1767146 8.599151e-01
> model  <- lm(sales ~ youtube + facebook, data = marketing)
> summary(model)

Call:
lm(formula = sales ~ youtube + facebook, data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.5572  -1.0502   0.2906   1.4049   3.3994 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.50532    0.35339   9.919   <2e-16 ***
youtube      0.04575    0.00139  32.909   <2e-16 ***
facebook     0.18799    0.00804  23.382   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.018 on 197 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8962 
F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
[redacted]
> confint(model)
                 2.5 %     97.5 %
(Intercept) 2.80841159 4.20222820
youtube     0.04301292 0.04849671
facebook    0.17213877 0.20384969
```