NAME:Akash Lalit mishra

ROLL NO:12

T.Y.B.Sc Computer Science

PRACTICAL

Data Science

CERTIFICATE

Jan Seva Sangh's

Shri Ram College Of Commerce

(Affiliated to the University Of Mumbai)

NAAC ACCREDITED 'B' GRADE (FIRST CYCLE)

University of Mumbai

CLASS: TYCS SUBJECT: Data Science SEAT NO/ROLL NO: 12

This is to certify that the work entered in this journal is the work of

Mr./Miss Akash Lalit Mishra

Who has worked for the practical examination of Data Science

Year B.S.C (CS) semester 6th of the year 2022-2023 in the college.

Internal |Signature External Signature

Date:

College Stamp Principal

INDEX

Practical-1

Aim: Use Google and Whois for Reconnaissance

Couchdb database ---

rscript Install

couchdb first

Rscript code

```
install.packages('sofa')
#devtools::install_github("rop ensci/sofa") library('sofa')
#create connection object x<-
Cushion$new()
#to check whether object created
x$ping() #create database ty db_create(x,dbna me = 'ty') db_list(x)
#create json doc
doc1<-'{"rollno":"01","name":"ABC","GRADE":"A"}'
doc_create(x,doc1,dbname = "ty",docid = "a_1")
doc2<-'{"rollno":"02","name":"PQR","GRADE":"A"}'
doc_create(x,doc2,dbname = "ty",docid = "a_2")
doc3<-'{"rollno":"03","name":"xyz","GRADE":"B","REMARK":"PASS"}'
doc_create(x,doc3,dbname = "ty",docid = "a_3")
#CHANGES FEED
db_changes(x,"ty")
#search for id > null so all docs will display
db_query(x,dbname = "ty", selector = list('_id'=list('$gt'=NULL)))$docs
#search for students with grade is A
db_query(x,dbname = "ty",selector = list(GRADE="A"))$docs
#search for students with remark =pass
db_query(x,dbname = "ty",selector = list(REMARK="PASS"))$docs
#return only certain fields where rollno>2
```

db_query(x,dbname = "ty",selector = list(rollno=list('$gt'='02')),fields=c("name","GRADE"))$docs

#convert the result of a query into a data frame using jsonlite library("jsonlite")

res<-db_query(x,dbname = "ty",selector = list('_id'=list('$gt'=NULL)),fields=c("name","rollno","GRADE","REMARK"),as="json")

#display json doc fromJSON(res)$docs

#doc_delete(cushion,d bname,docid)

doc_delete(x,dbname = "ty",docid = "a_2")

doc_get(x,dbname = "ty",docid = "a_2")

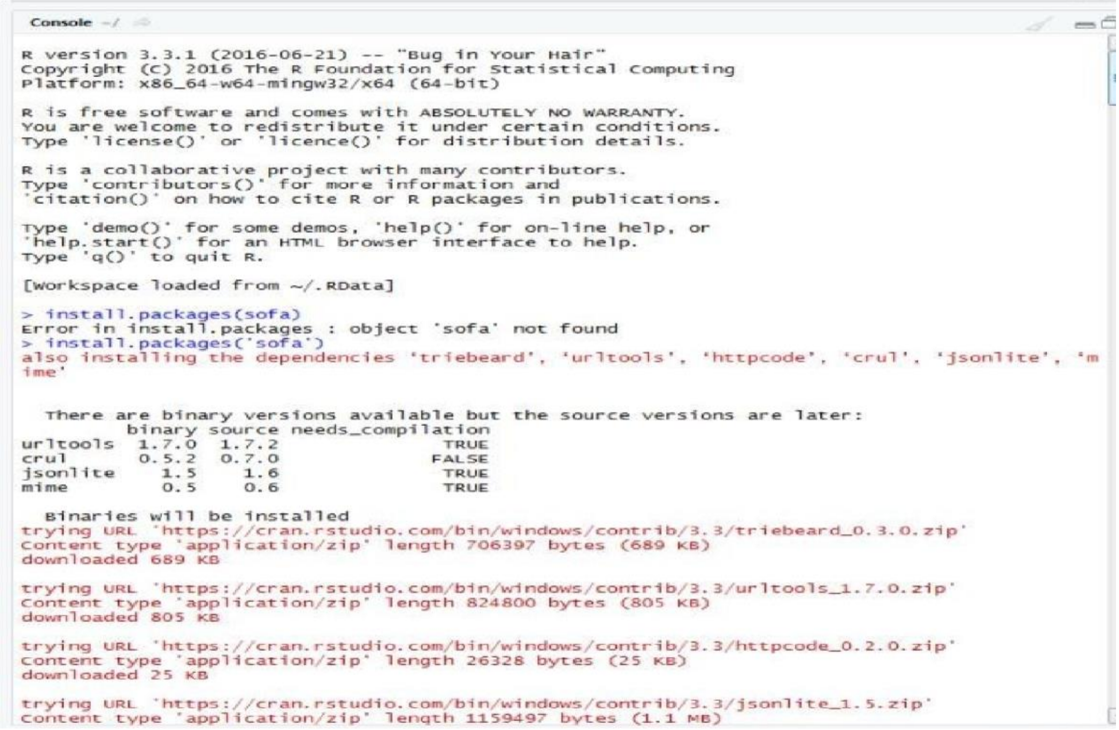
doc2<-'{"name":"Sdrink","beer":"TEST","note":"yummy","note2":"yay"}'

doc_update(x,dbname = "ty",doc=doc2,docid="a_3",rev = "3b1fb56db955b142c6efd3b3c52fe9e1b")

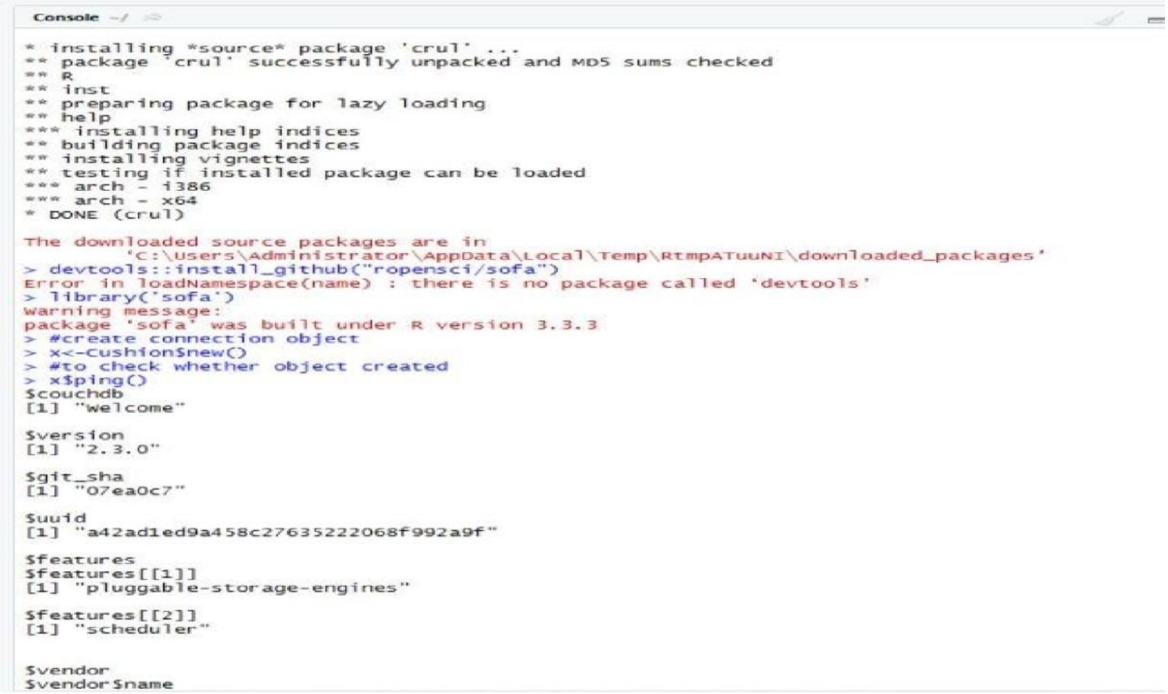
doc3<-'{"rollno":"01", "name":"UZMA","GRADE":"A"}'

doc_update(x,dbname = "ty",doc=doc3,docid = "a_1",rev = "1be7c98bddf8ea7c46f4f401ff387593d")

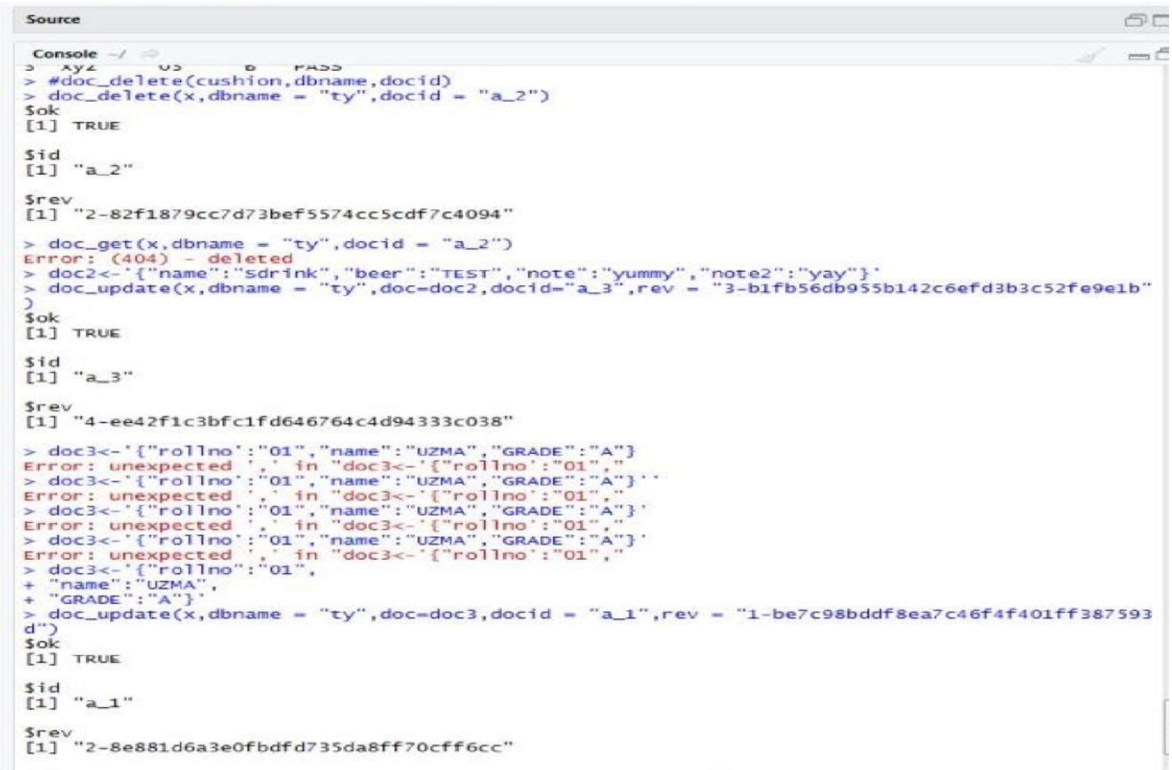
Output :



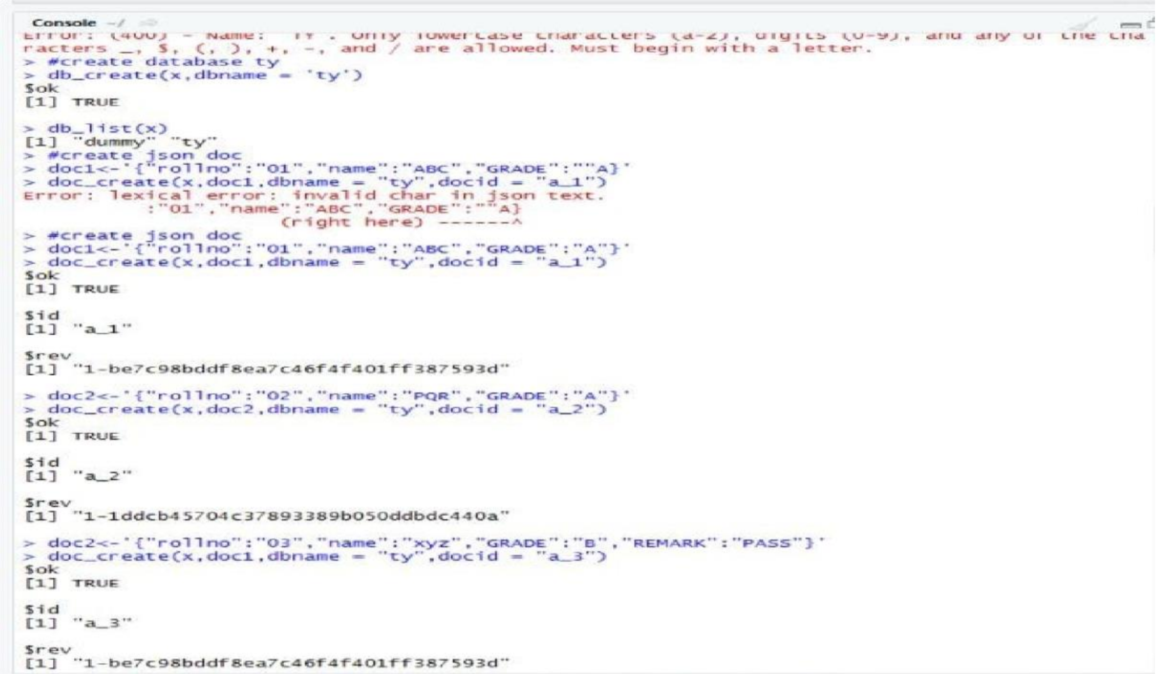
```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> install.packages(sofa)
Error in install.packages : object 'sofa' not found
> install.packages('sofa')
also installing the dependencies 'triebeard', 'urltools', 'httpcode', 'crul', 'jsonlite', 'm
ime'


  There are binary versions available but the source versions are later:
         binary source needs_compilation
urltools  1.7.0  1.7.2              TRUE
crul      0.5.2  0.7.0             FALSE
jsonlite    1.5    1.6              TRUE
mime        0.5    0.6              TRUE

  Binaries will be installed
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/triebeard_0.3.0.zip'
Content type 'application/zip' length 706397 bytes (689 KB)
downloaded 689 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/urltools_1.7.0.zip'
Content type 'application/zip' length 824800 bytes (805 KB)
downloaded 805 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/httpcode_0.2.0.zip'
Content type 'application/zip' length 26328 bytes (25 KB)
downloaded 25 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/jsonlite_1.5.zip'
Content type 'application/zip' length 1159497 bytes (1.1 MB)
```



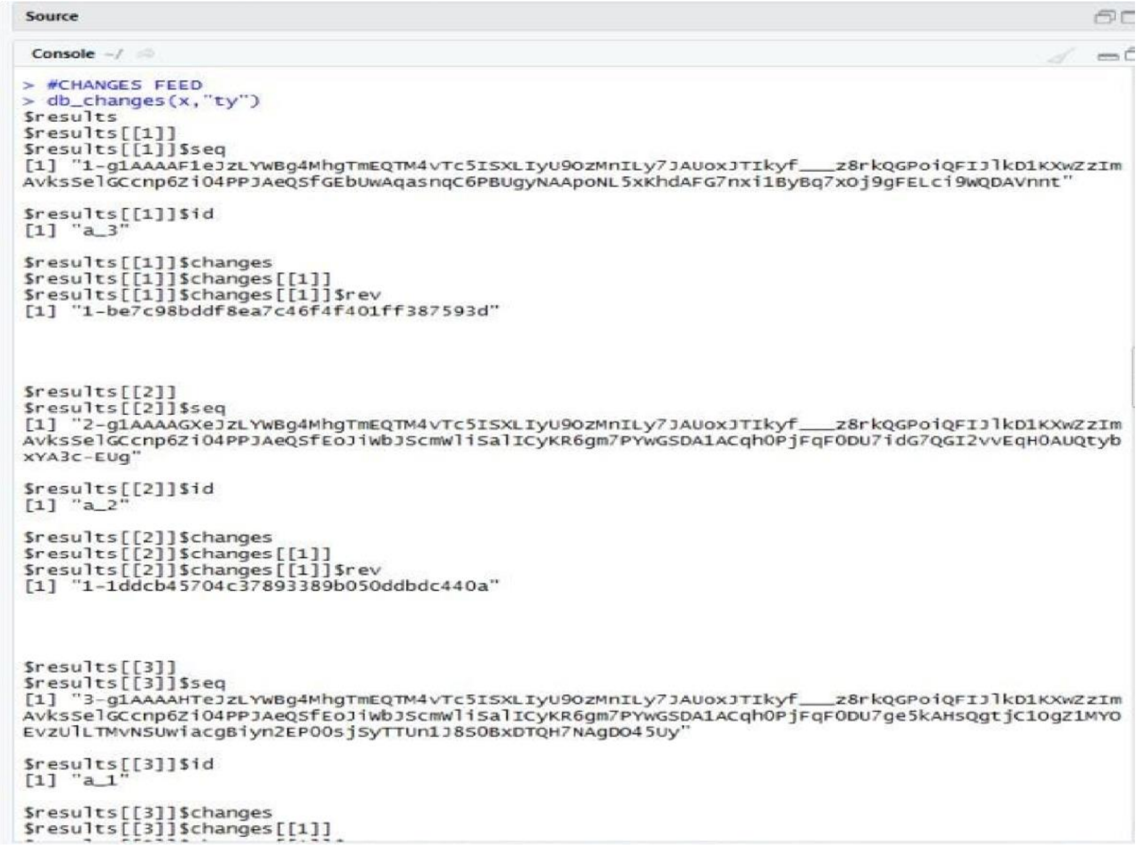
```
* installing *source* package 'crul' ...
** package 'crul' successfully unpacked and MD5 sums checked
** R
** inst
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded
*** arch - i386
*** arch - x64
* DONE (crul)

The downloaded source packages are in
        'C:\Users\Administrator\AppData\Local\Temp\RtmpATuuNI\downloaded_packages'
> devtools::install_github("ropensci/sofa")
Error in loadNamespace(name) : there is no package called 'devtools'
> library('sofa')
Warning message:
package 'sofa' was built under R version 3.3.3
> #create connection object
> x<-Cushion$new()
> #to check whether object created
> x$ping()
$couchdb
[1] "Welcome"

$version
[1] "2.3.0"

$git_sha
[1] "07ea0c7"

$uuid
[1] "a42ad1ed9a458c27635222068f992a9f"

$features
$features[[1]]
[1] "pluggable-storage-engines"

$features[[2]]
[1] "scheduler"


$vendor
$vendor$name
```



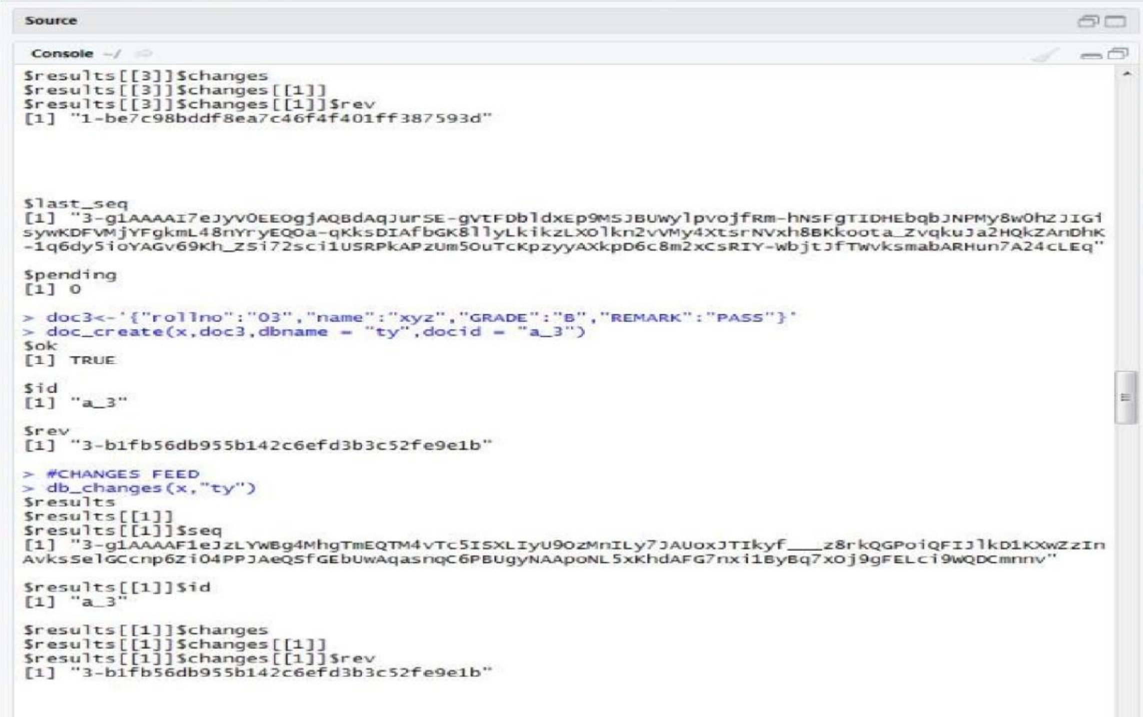
```
> xyz      05      6    PASS
> #doc_delete(cushion,dbname,docid)
> doc_delete(x,dbname = "ty",docid = "a_2")
$ok
[1] TRUE

$id
[1] "a_2"

$rev
[1] "2-82f1879cc7d73bef5574cc5cdf7c4094"

> doc_get(x,dbname = "ty",docid = "a_2")
Error: (404) - deleted
> doc2<-'{"name":"sdrink","beer":"TEST","note":"yummy","note2":"yay"}'
> doc_update(x,dbname = "ty",doc=doc2,docid="a_3",rev = "3-b1fb56db955b142c6efd3b3c52fe9e1b"
)
$ok
[1] TRUE

$id
[1] "a_3"

$rev
[1] "4-ee42f1c3bfc1fd646764c4d94333c038"

> doc3<-'{"rollno':"01","name":"UZMA","GRADE":"A"}
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno':"01","name":"UZMA","GRADE":"A"}''
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno':"01","name":"UZMA","GRADE":"A"}'
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno':"01","name":"UZMA","GRADE":"A"}'
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno":"01",
+ "name":"UZMA",
+ "GRADE":"A"}'
> doc_update(x,dbname = "ty",doc=doc3,docid = "a_1",rev = "1-be7c98bddf8ea7c46f4f401ff387593
d")
$ok
[1] TRUE

$id
[1] "a_1"

$rev
[1] "2-8e881d6a3e0fbdfd735da8ff70cff6cc"
```



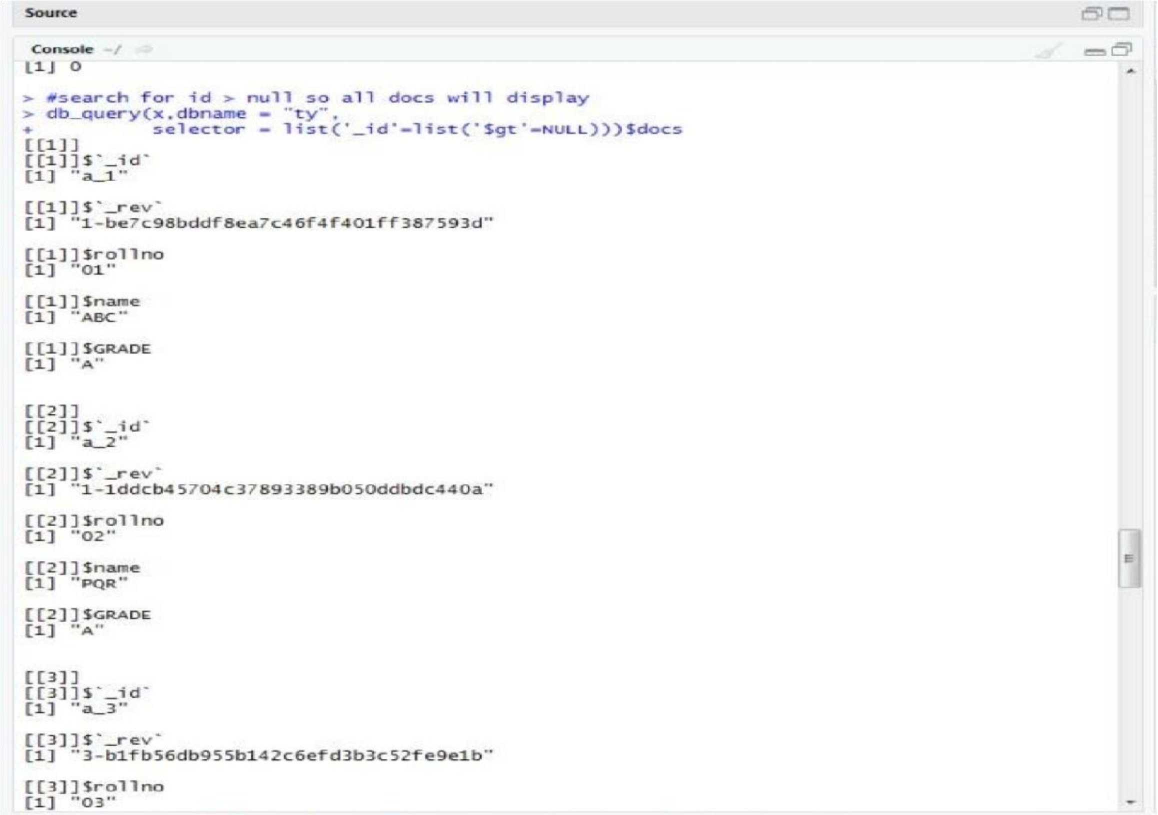
```
Error: (400) - Name: 'TY'. Only lowercase characters (a-z), digits (0-9), and any of the cha
racters _, $, (, ), +, -, and / are allowed. Must begin with a letter.
> #create database ty
> db_create(x,dbname = 'ty')
$ok
[1] TRUE

> db_list(x)
[1] "dummy" "ty"
> #create json doc
> doc1<-'{"rollno":"01","name":"ABC","GRADE":""A}'
> doc_create(x,doc1,dbname = "ty",docid = "a_1")
Error: lexical error: invalid char in json text.
          :"01","name":"ABC","GRADE":""A}
                     (right here) ------^
> #create json doc
> doc1<-'{"rollno":"01","name":"ABC","GRADE":"A"}'
> doc_create(x,doc1,dbname = "ty",docid = "a_1")
$ok
[1] TRUE

$id
[1] "a_1"

$rev
[1] "1-be7c98bddf8ea7c46f4f401ff387593d"

> doc2<-'{"rollno":"02","name":"PQR","GRADE":"A"}'
> doc_create(x,doc2,dbname = "ty",docid = "a_2")
$ok
[1] TRUE

$id
[1] "a_2"

$rev
[1] "1-1ddcb45704c37893389b050ddbdc440a"

> doc2<-'{"rollno":"03","name":"xyz","GRADE":"B","REMARK":"PASS"}'
> doc_create(x,doc1,dbname = "ty",docid = "a_3")
$ok
[1] TRUE

$id
[1] "a_3"

$rev
[1] "1-be7c98bddf8ea7c46f4f401ff387593d"
```



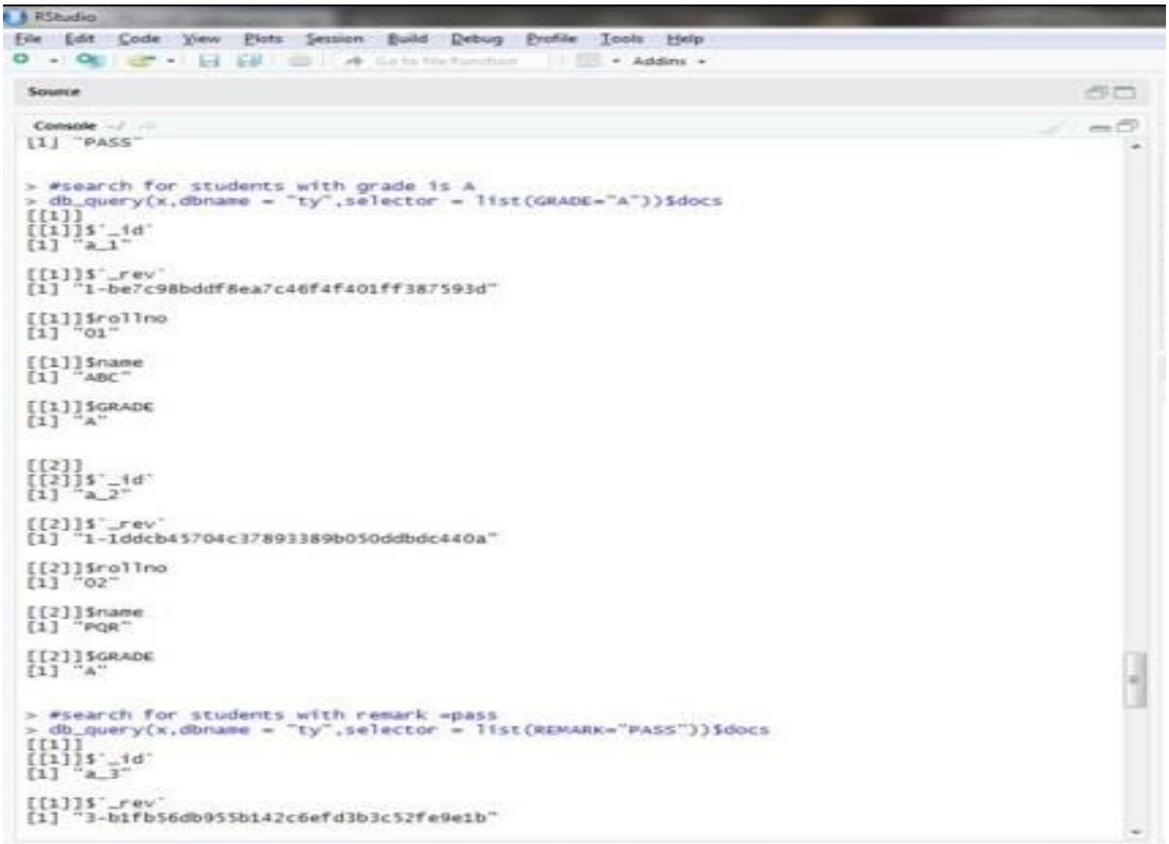
```
> #CHANGES FEED
> db_changes(x,"ty")
$results
$results[[1]]
$results[[1]]$seq
[1] "1-g1AAAAF1eJzLYWBg4MhgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUoxJTIkyf___z8rkQGPoiQFIJlkD1KXwZzIm
AvksSelGCcnp6Zi04PPJAeQSfGEbUwAqasnqC6PBUgyNAApoNL5xKhdAFG7nxi1ByBq7xOj9gFELci9WQDAVnnt"

$results[[1]]$id
[1] "a_3"

$results[[1]]$changes
$results[[1]]$changes[[1]]
$results[[1]]$changes[[1]]$rev
[1] "1-be7c98bddf8ea7c46f4f401ff387593d"



$results[[2]]
$results[[2]]$seq
[1] "2-g1AAAAGXeJzLYWBg4MhgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUoxJTIkyf___z8rkQGPoiQFIJlkD1KXwZzIm
AvksSelGCcnp6Zi04PPJAeQSfEoJiwbJScmWliSalICyKR6gm7PYWGSDA1ACqh0PjFqF0DU7idG7QGI2vvEqH0AUQtyb
xYA3c-EUg"

$results[[2]]$id
[1] "a_2"

$results[[2]]$changes
$results[[2]]$changes[[1]]
$results[[2]]$changes[[1]]$rev
[1] "1-1ddcb45704c37893389b050ddbdc440a"



$results[[3]]
$results[[3]]$seq
[1] "3-g1AAAAHTeJzLYWBg4MhgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUoxJTIkyf___z8rkQGPoiQFIJlkD1KXwZzIm
AvksSelGCcnp6Zi04PPJAeQSfEoJiwbJScmWliSalICyKR6gm7PYWGSDA1ACqh0PjFqF0DU7ge5kAHSQgtjC1OgZ1MYO
EvzUlLTMvNSUwiacgBiyn2EP00sjSyTTUn1J8S0BxDTQH7NAgDO45Uy"

$results[[3]]$id
[1] "a_1"

$results[[3]]$changes
$results[[3]]$changes[[1]]
```



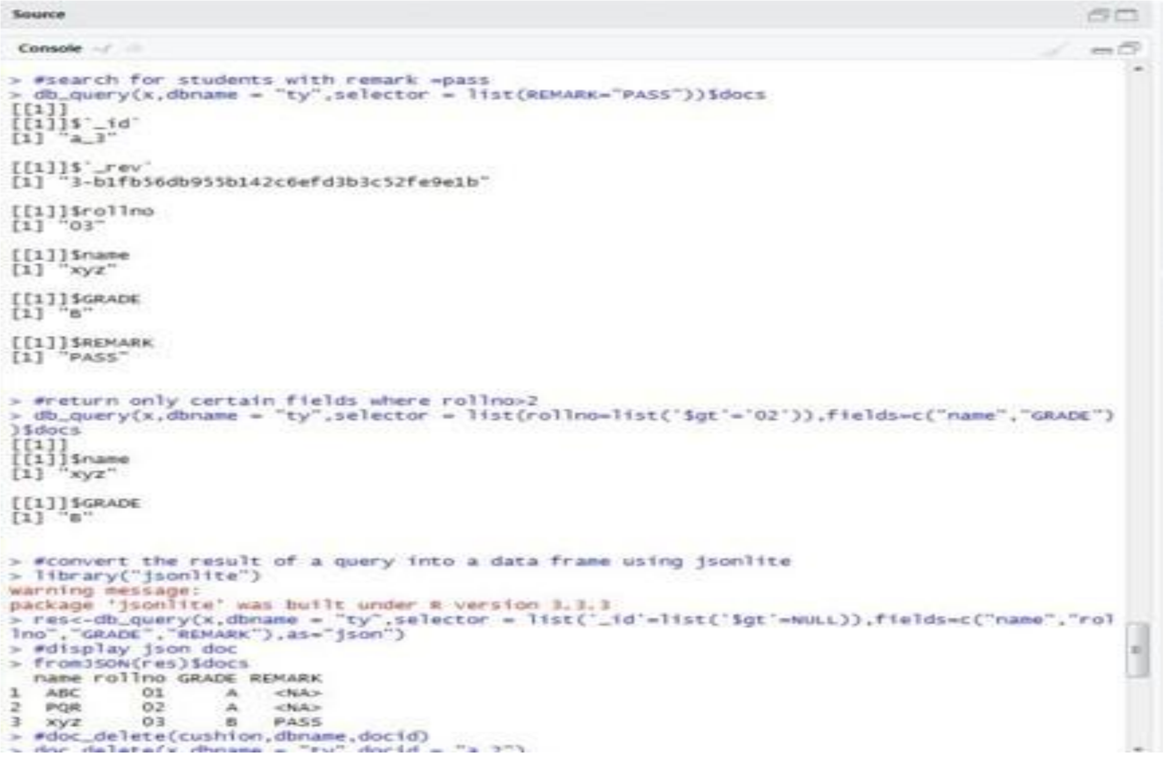
```
$results[[3]]$changes
$results[[3]]$changes[[1]]
$results[[3]]$changes[[1]]$rev
[1] "1-be7c98bddf8ea7c46f4f401ff387593d"




$last_seq
[1] "3-g1AAAAI7eJyV0EEOgjAQBdAqJurSE-gVtFDbldxEp9MSJBUWylpvojfRm-hNSFgTIDHEbqbJNPMy8w0hZJIGi
sywKDFVMjYFgkmL48nYryEQ0a-qKksDIAfbGK8llyLkikzLXOlkn2vVMy4XtsrNVxh8BKkoota_ZvqkuJa2HQkZAnDhK
-1q6dy5ioYAGv69Kh_ZSi72sci1USRPkAPzUm5OuTcKpzyyAXkpD6c8m2xCsRIY-WbjtJfTWvksmabARHun7A24cLEq"

$pending
[1] 0

> doc3<-'{"rollno":"03","name":"xyz","GRADE":"B","REMARK":"PASS"}'
> doc_create(x,doc3,dbname = "ty",docid = "a_3")
$ok
[1] TRUE

$id
[1] "a_3"

$rev
[1] "3-b1fb56db955b142c6efd3b3c52fe9e1b"

> #CHANGES FEED
> db_changes(x,"ty")
$results
$results[[1]]
$results[[1]]$seq
[1] "3-g1AAAAF1eJzLYWBg4MhgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUoxJTIkyf___z8rkQGPoiQFIJlkD1KXwZzIn
AvksSelGCcnp6Zi04PPJAeQSfGEbUwAqasnqC6PBUgyNAApoNL5xKhdAFG7nxi1ByBq7xOj9gFELci9WQDCmnnv"

$results[[1]]$id
[1] "a_3"

$results[[1]]$changes
$results[[1]]$changes[[1]]
$results[[1]]$changes[[1]]$rev
[1] "3-b1fb56db955b142c6efd3b3c52fe9e1b"
```



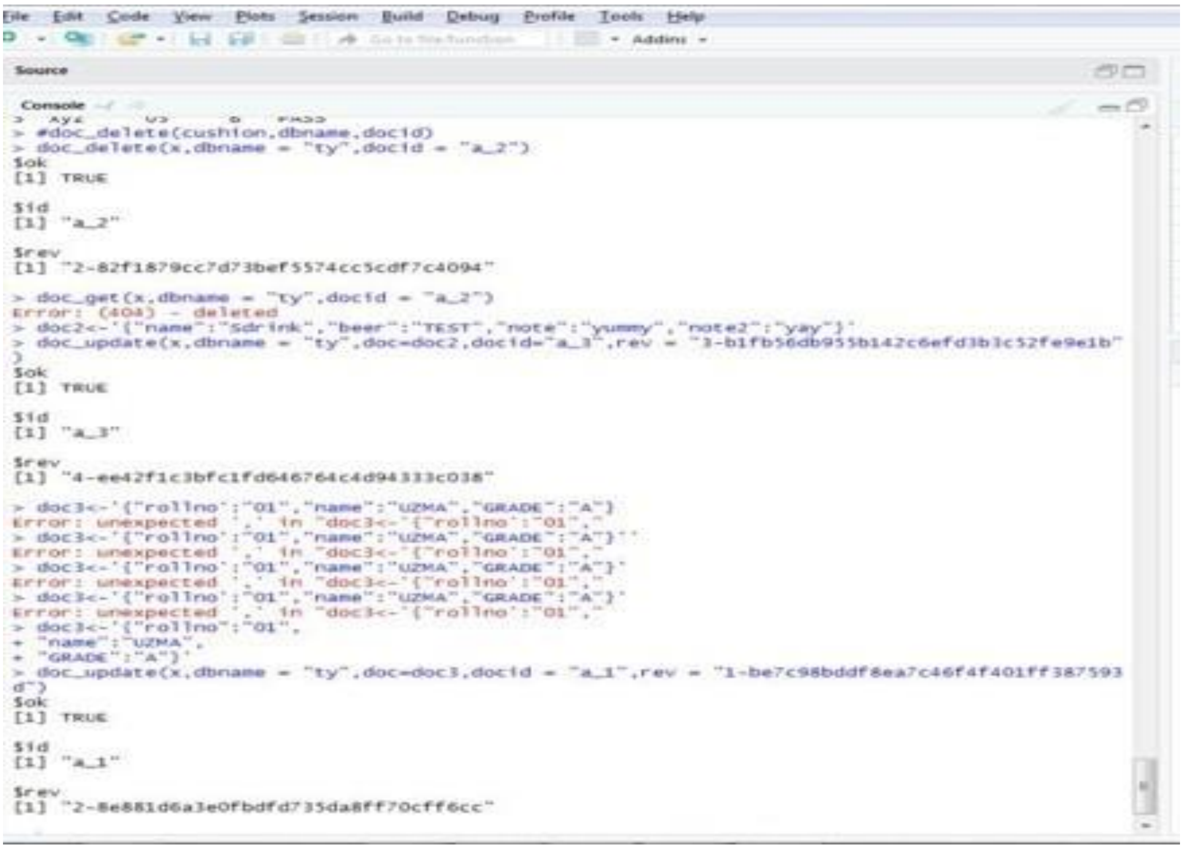
```
[1] 0

> #search for id > null so all docs will display
> db_query(x,dbname = "ty",
+          selector = list('_id'=list('$gt'=NULL)))$docs
[[1]]
[[1]]$`_id`
[1] "a_1"

[[1]]$`_rev`
[1] "1-be7c98bddf8ea7c46f4f401ff387593d"

[[1]]$rollno
[1] "01"

[[1]]$name
[1] "ABC"

[[1]]$GRADE
[1] "A"


[[2]]
[[2]]$`_id`
[1] "a_2"

[[2]]$`_rev`
[1] "1-1ddcb45704c37893389b050ddbdc440a"

[[2]]$rollno
[1] "02"

[[2]]$name
[1] "PQR"

[[2]]$GRADE
[1] "A"


[[3]]
[[3]]$`_id`
[1] "a_3"

[[3]]$`_rev`
[1] "3-b1fb56db955b142c6efd3b3c52fe9e1b"

[[3]]$rollno
[1] "03"
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console ~/

```
[1] "PASS"


> #search for students with grade is A
> db_query(x,dbname = "ty",selector = list(GRADE="A"))$docs
[[1]]
[[1]]$`_id`
[1] "a_1"

[[1]]$`_rev`
[1] "1-be7c98bddf8ea7c46f4f401ff387593d"

[[1]]$rollno
[1] "01"

[[1]]$name
[1] "ABC"

[[1]]$GRADE
[1] "A"


[[2]]
[[2]]$`_id`
[1] "a_2"

[[2]]$`_rev`
[1] "1-1ddcb45704c37893389b050ddbdc440a"

[[2]]$rollno
[1] "02"

[[2]]$name
[1] "PQR"

[[2]]$GRADE
[1] "A"


> #search for students with remark =pass
> db_query(x,dbname = "ty",selector = list(REMARK="PASS"))$docs
[[1]]
[[1]]$`_id`
[1] "a_3"

[[1]]$`_rev`
[1] "3-b1fb56db955b142c6efd3b3c52fe9e1b"
```



```
> #search for students with remark =pass
> db_query(x,dbname = "ty",selector = list(REMARK="PASS"))$docs
[[1]]
[[1]]$`_id`
[1] "a_3"

[[1]]$`_rev`
[1] "3-b1fb56db955b142c6efd3b3c52fe9e1b"

[[1]]$rollno
[1] "03"

[[1]]$name
[1] "xyz"

[[1]]$GRADE
[1] "B"

[[1]]$REMARK
[1] "PASS"


> #return only certain fields where rollno>2
> db_query(x,dbname = "ty",selector = list(rollno=list('$gt'='02')),fields=c("name","GRADE")
)$docs
[[1]]
[[1]]$name
[1] "xyz"

[[1]]$GRADE
[1] "B"


> #convert the result of a query into a data frame using jsonlite
> library("jsonlite")
warning message:
package 'jsonlite' was built under R version 3.3.3
> res<-db_query(x,dbname = "ty",selector = list('_id'=list('$gt'=NULL)),fields=c("name","rol
lno","GRADE","REMARK"),as="json")
> #display json doc
> fromJSON(res)$docs
  name rollno GRADE REMARK
1  ABC     01     A   <NA>
2  PQR     02     A   <NA>
3  xyz     03     B   PASS
> #doc_delete(cushion,dbname,docid)
> doc_delete(x,dbname = "ty",docid = "a_2")
```


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console

```
3  xyz     03     B   PASS
> #doc_delete(cushion,dbname,docid)
> doc_delete(x,dbname = "ty",docid = "a_2")
$ok
[1] TRUE

$id
[1] "a_2"

$rev
[1] "2-82f1879cc7d73bef5574cc5cdf7c4094"

> doc_get(x,dbname = "ty",docid = "a_2")
Error: (404) - deleted
> doc2<-'{"name":"sdrink","beer":"TEST","note":"yummy","note2":"yay"}'
> doc_update(x,dbname = "ty",doc=doc2,docid="a_3",rev = "3-b1fb56db955b142c6efd3b3c52fe9e1b"
)
$ok
[1] TRUE

$id
[1] "a_3"

$rev
[1] "4-ee42f1c3bfc1fd646764c4d94333c038"

> doc3<-'{"rollno":"01","name":"UZMA","GRADE":"A"}
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno':"01","name":"UZMA","GRADE":"A"}''
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno':"01","name":"UZMA","GRADE":"A"}'
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno':"01","name":"UZMA","GRADE":"A"}'
Error: unexpected ',' in "doc3<-'{"rollno':"01","
> doc3<-'{"rollno":"01",
+ "name":"UZMA",
+ "GRADE":"A"}'
> doc_update(x,dbname = "ty",doc=doc3,docid = "a_1",rev = "1-be7c98bddf8ea7c46f4f401ff387593
d")
$ok
[1] TRUE

$id
[1] "a_1"

$rev
[1] "2-8e881d6a3e0fbdfd735da8ff70cff6cc"
```

Practical-2

Aim: Practical of Data collection, Data curation and management for Large-scale Data system (such as MongoDB)

- MongoDB Create database
- MongoDB Drop Database
- MongoDB Create collection
- MongoDB Drop collection
- MongoDB Insert Document
- MongoDB Query Document
- MongoDB Update Document
- Delete document in MongoDB
- MongoDB Projection
- limit() and skip() method in MongoDB
- Sorting of Documents in MongoDB
- MongoDB Indexing

Starting server with mongo or mongodb

C:\>**mongo**

>db

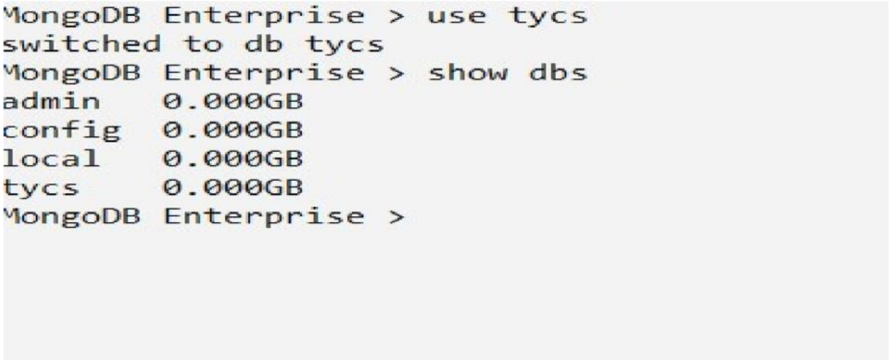
Test

- ## Create Database in MongoDB

Once you are in the MongoDB shell, create the database in MongoDB by typing this command: use database_name

For example: create a database “tycs”:

> use tycs switched to db tycs



```
MongoDB Enterprise > use tycs
switched to db tycs
MongoDB Enterprise > show dbs
admin    0.000GB
config   0.000GB
local    0.000GB
tycs     0.000GB
MongoDB Enterprise >
```

create a collection **user** and insert a document in it. > db.user.insert({name: "Asif", age: 20})

O/P: WriteResult({ "nInserted" : 1 })

>show dbs admin 0.000GB

config 0.000GB local 0.000GB

tycs 0.000GB

MongoDB Drop Database

The syntax to drop a Database is:

>db.dropDatabase()

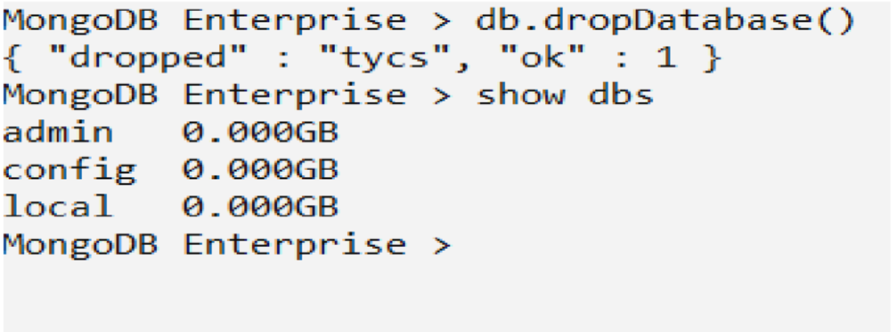
O/P:

{ "dropped" : "Testdb", "ok" : 1 } MongoDB Enterprise

> show dbs admin 0.000GB config 0.000GB local

0.000GB tycs 0.000GB

O/P:



```
MongoDB Enterprise > db.dropDatabase()
{ "dropped" : "tycs", "ok" : 1 }
MongoDB Enterprise > show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
MongoDB Enterprise >
```

Create Collection in MongoDB

Method 1: Creating the Collection in MongoDB on the fly

MongoDB Enterprise > use tycs switched to db tycs

MongoDB Enterprise > db.tycs.insert({name:"Asif khan",age:21,website:"www.google.com"}) **O/P:**

WriteResult({ "nInserted" : 1 })

Syntax: **db.collection_name.find()**

```
MongoDB Enterprise > db.tycs.find()
```

o/p:

```
{ "_id" : ObjectId("5e410808e3755b1e06a63d1d"), "name" : "Asif khan",

"age" : 21, "website" : "www.google.com" }
```

```
show collections
```

```
MongoDB Enterprise > show collections
```

O/P: tycs user

- **Drop collection in MongoDB**

SYNTAX:

```
db.collection_name.drop()
```

```
MongoDB Enterprise > use students switched to db students

MongoDB Enterprise > show collections students teachers
tycs user

MongoDB Enterprise > db.user.drop() true

MongoDB Enterprise > show collections students teacher
tycs
```

MongoDB Insert Document

Syntax to insert a document into the collection:

```
db.collection_name.insert()
```

```
> db.tycs.insert(
...     {
```

```
...      name: "ASIF",
...      age: 20,
...      email: "asif@gmail.com",
...      course: [ { name: "MongoDB", duration: 7 }, { name: "Java", duration: 30 } ]
...    } ... )
```

O/P:

```
WriteResult({ "nInserted" : 1 })
```

Verification: Syntax:

```
db .collection_name    .find ()
```

```
> db.tycs.find()
```

```
{ "_id" : ObjectId("5c2d37734fa204bd77e7fc1c"), "name" : "ASIF", "age" : 20, "email" : "asif@gmail.com",
"course" : [ { "name" :

"MongoDB", "duration" : 7 }, { "name" : "Java", "duration" : 30 } ] }
```

MongoDB Example: Insert Multiple Documents in collection

```
MongoDB Enterprise > var beginners=

... [

... "studentID":1001,

... "studentName":"Asif",

... "age":20

... },

... ]
```

- **MongoDB Query Document using find() method**

Querying all the documents in JSON format

```
MongoDB Enterprise > db.students.find().pretty()

{
    "_id" : ObjectId("5e410f3fe3755b1e06a63d1e"),

    "studentID" : 1001,

    "studentName" : "Asif",

    "age" : 20

}
```

- **Query Document based on the criteria**

```
> db.students.find({StudentName : "Asif"}).pretty()

{

    "_id" : ObjectId("5c281c90c23e08d1515fd9cc"),

    "StudentId" : 1001,

    "StudentName" : "Asif",

    "age" : 20

}
```

- **Updating Document using update() method**

Syntax:

```
db.collection_name.update(criteria, update_data)
```

```
> use tycs switched to db tycs >
show collections
beginnersbook students tycs

> db.createCollection("got")

{ "ok" : 1 }

> var abc = [

... {

...     "_id" : ObjectId("59bd2e73ce524b733f14dd65"),

...     "name" : "Asif",
...       "age" : 20
... },
... ];
> db.got.find().pretty()
{
        "_id" : ObjectId("59bd2e73ce524b733f14dd65"),
        "name" : "steve",
        "age" : 20
}
```

To update multiple documents with the update() method:

```
db.got.update({"name":"Jon Snow"},  {$set:{"name":"Kit Harington"}},{multi:true})
```

Updating Document using save() method **Syntax:** db.collection_name.save(

```
{_id:ObjectId(), new_document} )
```

To get the _id of a document, you can either type this command:

```
db.got.find().pretty()
```

```
> db.got.find({"name": "Asif"}).pretty()
{
        "_id" : ObjectId("59bd2e73ce524b733f14dd65"),
        "name" : "Asif",
        "age" : 20
}
> db.got.find().pretty()
{
        "_id" : ObjectId("59bd2e73ce524b733f14dd65"),

        "name" : "Steve",

        "age" : 20

}
```

- **MongoDB Delete Document from a Collection Syntax of remove()**

method: db.collection_name.remove(delete_criteria)

Delete Document using remove() method

```
> db.students.find().pretty          ()
{
        "_id"  :  ObjectId    ("59bcecc7668dcce02aaa6fed"           ),
        "StudentId"     :  1001   ,
        "StudentName"      :  "Steve"    ,
        "age"   :   30
}
```

```
db.students.remove({"StudentId": 3333})
```

Output:

```
WriteResult({ "nRemoved" : 1 })
```

To verify whether the document is actually deleted. Type the following command:

```
db.students.find().pretty()
```

It will list all the documents of students collection.

```
> use tycs switched to db  tycs >
db.students.find().pretty()

{

        "_id" : ObjectId("5c281c90c23e08d1515fd9cc"),

        "StudentId" : 1001,

        "StudentName" : "Asif",

        "age" : 20

}

{
        "_id" : ObjectId("5c2d38934fa204bd77e7fc1d"),

        "StudentId" : 1001,

        "StudentName" : "Steve",

        "age" : 30

}
```

Remove all Documents

```
db.collection_name.remove({})
```

- **MongoDB Projection Syntax:**

```
db.collection_name.find({},{field_key:1 or 0})
```

```
> db.students.find().pretty()

{

    "_id" : ObjectId("5c281c90c23e08d1515fd9cc"),

    "StudentId" : 1001,

    "StudentName" : "Steve",

    "age" : 20

}

> db.students.find({}, {"_id": 0, "StudentId" : 1})

{ "StudentId" : 1001 }

{ "StudentId" : 1002 }


> db.students.find({}, {"_id": 0, "StudentName" : 0, "age" : 0})
```

{ "StudentId" : 1001 }

{ "StudentId" : 1002 }

- **MongoDB – limit( ) and skip( ) method**

The **limit() method in MongoDB**

Syntax:

```
db.collection_name.find().limit(number_of_documents) db.studentdata.find({student_id : {$gt:2002}}).pretty()
db.studentdata.find({student_id : {$gt:2002}}).limit(1).pretty()
```

MongoDB Skip() Method

```
db.studentdata.find({student_id : {$gt:2002}}).limit(1).skip(1).pretty()
```

- **MongoDB sort() method**

Sorting Documents using sort() method

Syntax of sort() method:

```
db.collecttion_name.find().sort({field_key:1 or -1})
```

1 is for ascending order and -1 is for descending order. The default value is 1.

For example: collection studentdata contains following documents:

```
> db.studentdata.find().pretty()

{

      "_id" : ObjectId("59bf63380be1d7770c3982af"),

      "student_name" : "Steve",

      "student_id" : 1001,

      "student_age" :1002

}
```

Let's display the student_id of all the documents in **descending order**:

> db.studentdata.find({}, {"student_id": 1, _id:0}).sort({"student_id": -1})

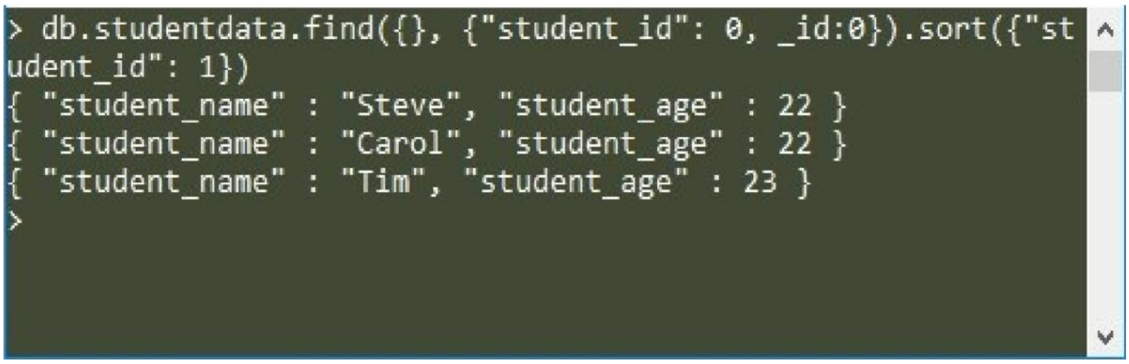
{ "student_id" : 1001 }
{ "student_id" : 1002 }

To display the student_id field of all the students in **ascending order**:

> db.studentdata.find({}, {"student_id": 1, _id:0}).sort({"student_id": 1})

{ "student_id" : 1001 }

{ "student_id" : 1002 }



```
> db.studentdata.find({}, {"student_id": 0, _id:0}).sort({"st
udent_id": 1})
{ "student_name" : "Steve", "student_age" : 22 }
{ "student_name" : "Carol", "student_age" : 22 }
{ "student_name" : "Tim", "student_age" : 23 }
>
```

- **MongoDB Indexing with Example How to create index in MongoDB**

db.collection_name.createIndex({field_name: 1 or -1})

The value 1 is for ascending order and -1 is for **descending order.**

Let's create the index on student_name field in **ascending order:**

db.studentdata.createIndex({student_name: 1}) **Output:**

```
{
        "createdCollectionAutomatically"            : false   ,
        "numIndexesBefore"          : 1 ,
        "numIndexesAfter"          : 2 ,
        "ok"   : 1
}
```

- **MongoDB – Finding the indexes in a collection**

```
db.collection_name.getIndexes()

> db.studentdata.getIndexes()

[
```

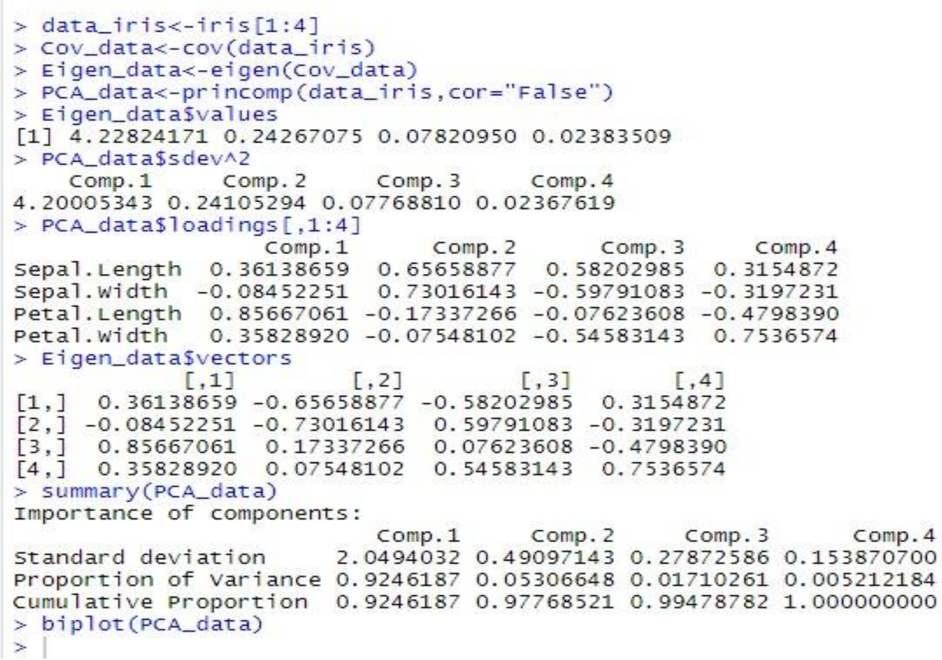
```
        {
                "v"  : 2 ,
                "key"   : {
                        "_id"   : 1
                },
                "name"    : "_id_"  ,
                "ns"   : "test.studentdata"
        },
]
```

Practical-3

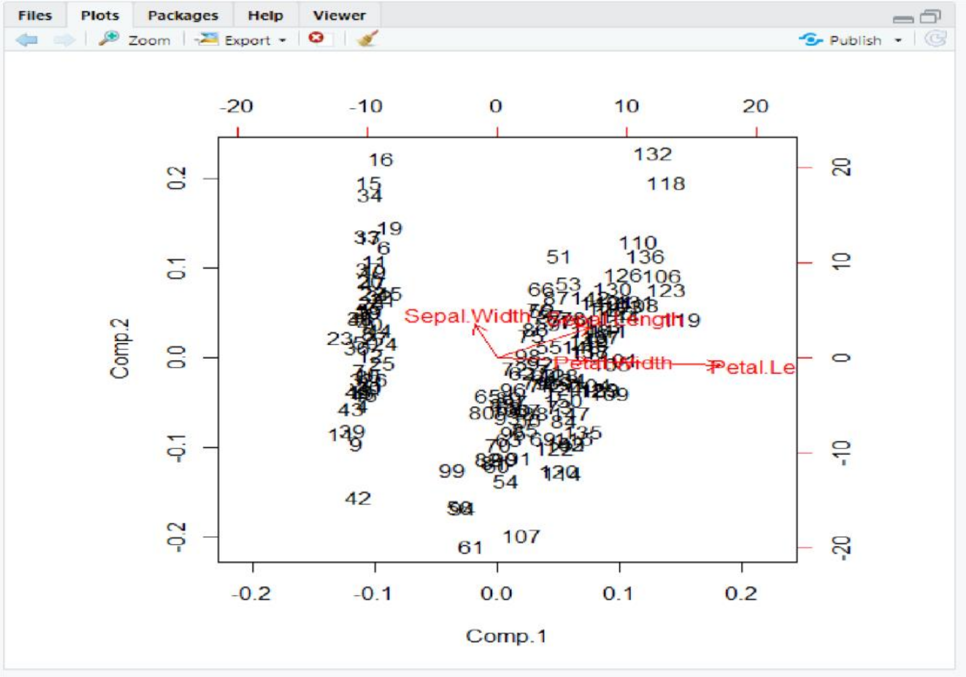
Aim: Practical of Principal Component Analysis

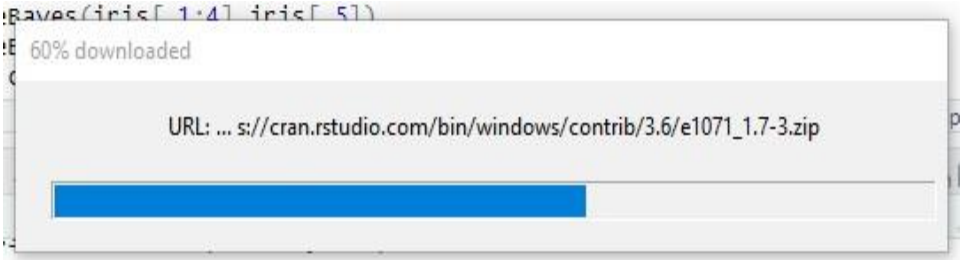
```
data_iris<-iris[1:4] Cov_data<-cov(data_iris)

Eigen_data<-eigen(Cov_data)

PCA_data<-princomp(data_iris,cor="False")

Eigen_data$values

PCA_data$sdev^2

PCA_data$loadings[,1:4] Eigen_data$vectors summary(PCA_data)

biplot(PCA_data)  screeplot(PCA_data,type="lines")

model2=PCA_data$loadings[,1] model2_scores<-

as.matrix(data_iris)%*%model2 library(class) install.packages("e1071")

library(e1071)

mod1<-naiveBayes(iris[,1:4],iris[,5]) mod2<-naiveBayes(model2_scores,iris[,5])

table(predict(mod1,iris[,1:4]),iris[,5]) table(predict(mod2,model2_scores),iris[,5])
```

Output :

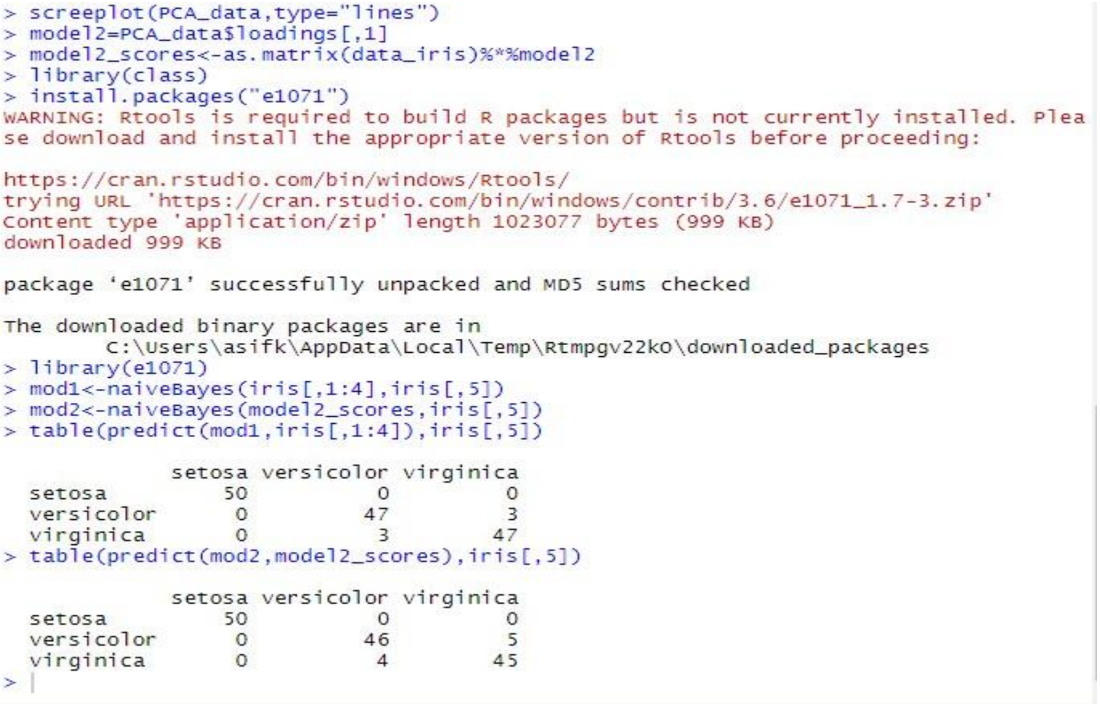


```
> data_iris<-iris[1:4]
> Cov_data<-cov(data_iris)
> Eigen_data<-eigen(Cov_data)
> PCA_data<-princomp(data_iris,cor="False")
> Eigen_data$values
[1] 4.22824171 0.24267075 0.07820950 0.02383509
> PCA_data$sdev^2
    Comp.1     Comp.2     Comp.3     Comp.4 
4.20005343 0.24105294 0.07768810 0.02367619 
> PCA_data$loadings[,1:4]
                  Comp.1      Comp.2      Comp.3     Comp.4
Sepal.Length  0.36138659  0.65658877  0.58202985  0.3154872
Sepal.Width  -0.08452251  0.73016143 -0.59791083 -0.3197231
Petal.Length  0.85667061 -0.17337266 -0.07623608 -0.4798390
Petal.Width   0.35828920 -0.07548102 -0.54583143  0.7536574
> Eigen_data$vectors
            [,1]        [,2]        [,3]       [,4]
[1,]  0.36138659 -0.65658877 -0.58202985  0.3154872
[2,] -0.08452251 -0.73016143  0.59791083 -0.3197231
[3,]  0.85667061  0.17337266  0.07623608 -0.4798390
[4,]  0.35828920  0.07548102  0.54583143  0.7536574
> summary(PCA_data)
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000
> biplot(PCA_data)
> 
```

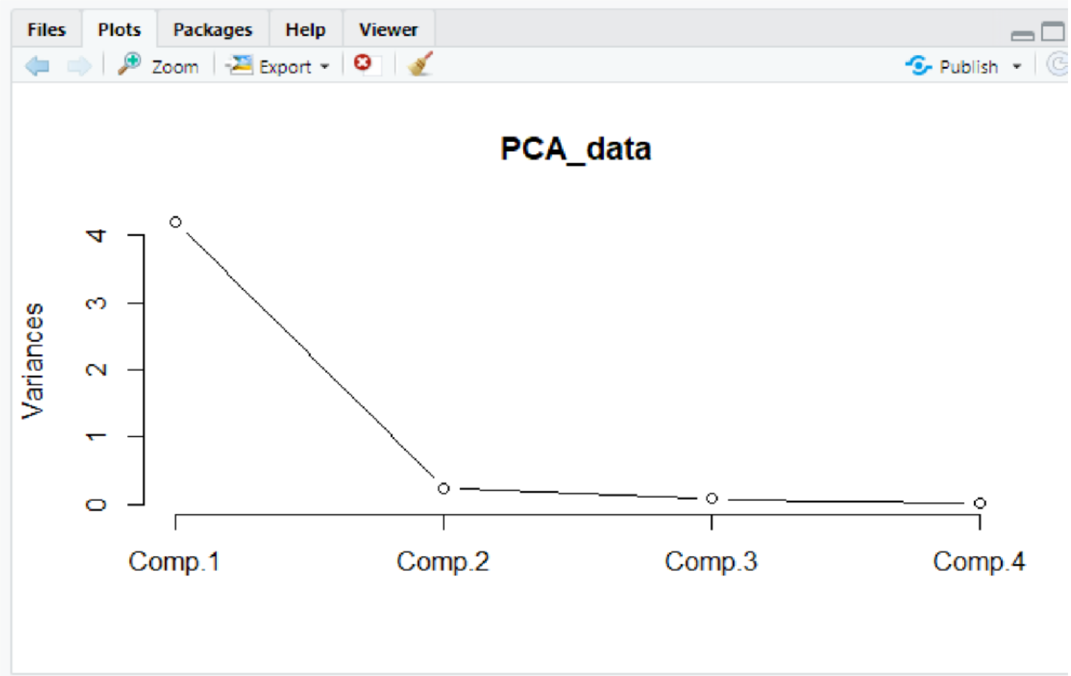




URL: ... s://cran.rstudio.com/bin/windows/contrib/3.6/e1071_1.7-3.zip



```
> screeplot(PCA_data,type="lines")
> model2=PCA_data$loadings[,1]
> model2_scores<-as.matrix(data_iris)%*%model2
> library(class)
> install.packages("e1071")
WARNING: Rtools is required to build R packages but is not currently installed. Plea
se download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/e1071_1.7-3.zip'
Content type 'application/zip' length 1023077 bytes (999 KB)
downloaded 999 KB

package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asifk\AppData\Local\Temp\Rtmpgv22kO\downloaded_packages
> library(e1071)
> mod1<-naiveBayes(iris[,1:4],iris[,5])
> mod2<-naiveBayes(model2_scores,iris[,5])
> table(predict(mod1,iris[,1:4]),iris[,5])

             setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         47         3
  virginica       0          3        47
> table(predict(mod2,model2_scores),iris[,5])

             setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         46         5
  virginica       0          4        45
> |
```

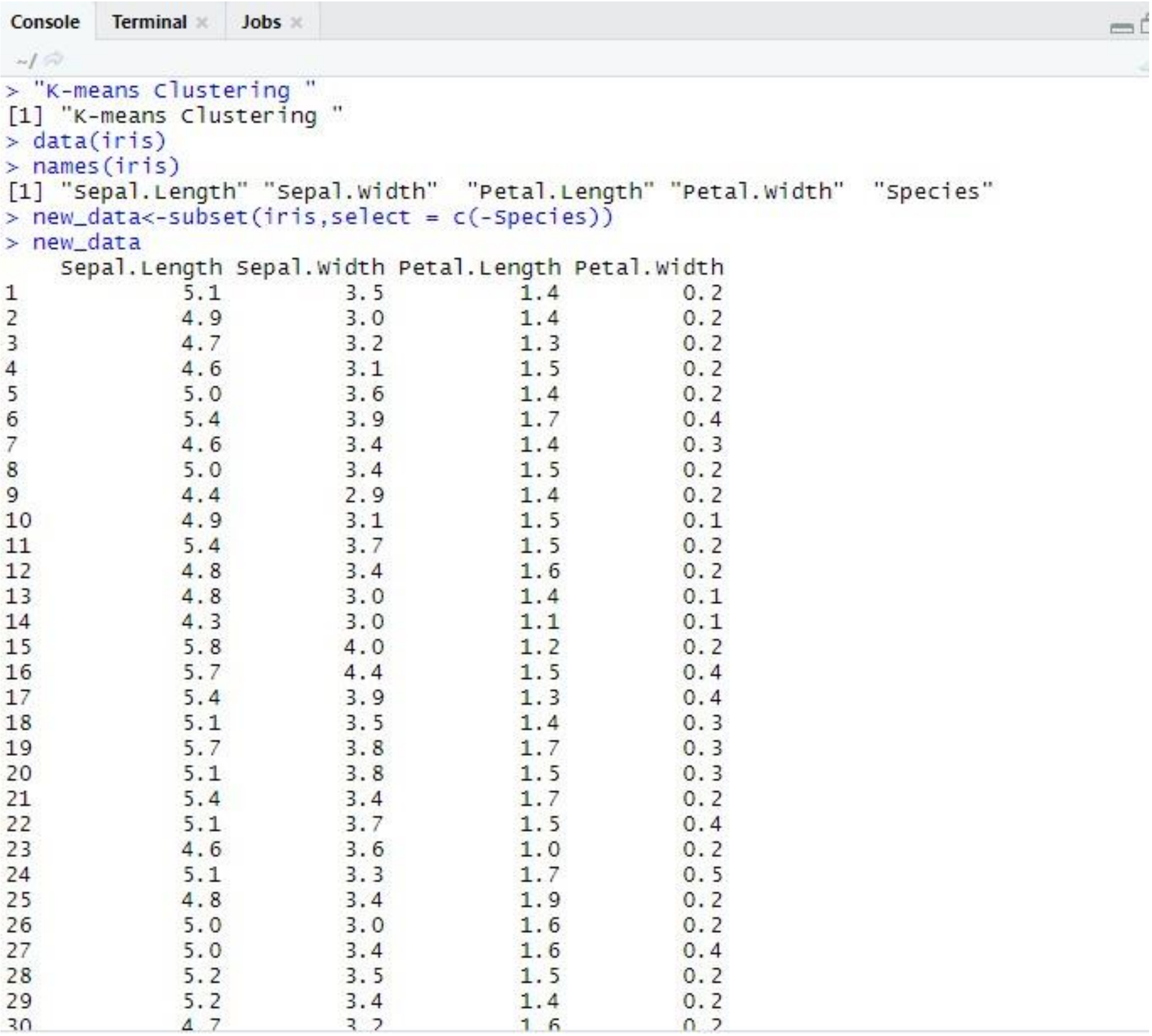

Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
PCA_data
Variances
0
1
2
3
4
Comp.1
Comp.2
Comp.3
Comp.4

Practical-4

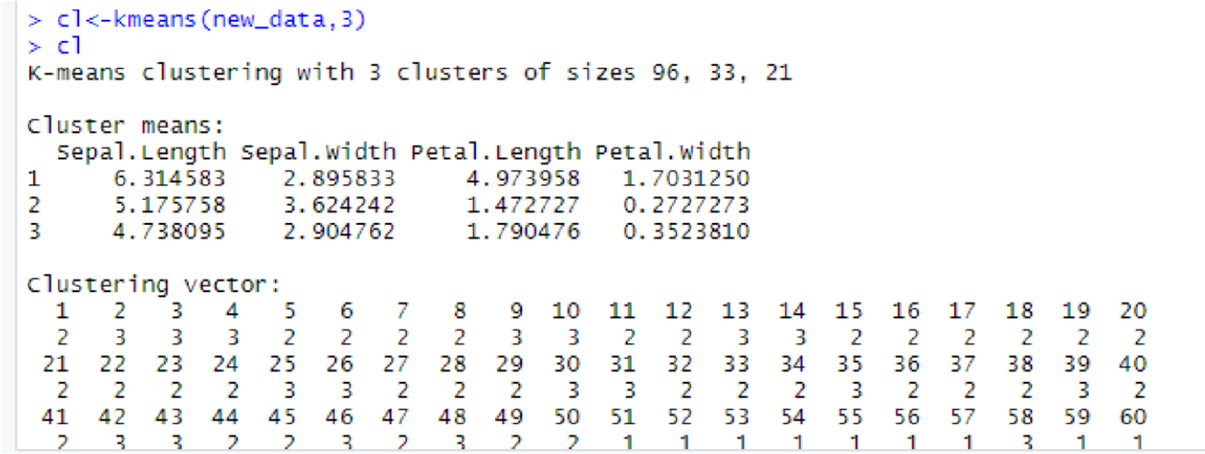
Aim: Practical of Clustering

```
"K-means Clustering " data(iris) names(iris) new_data<-
subset(iris,select = c(-Species)) new_data cl<-kmeans(new_data,3)
cl
data<-new_data wss<-sapply(1:15,function(k){kmeans(data,k)$tot.withinss}) wss
plot(1:15,wss,type="b",pch=19,frame=FALSE,xlab ="Number of clusters K",ylab =
"Total within-clusters sums of squares") library(cluster)
clusplot(new_data,cl$cluster,color=TRUE,shade=TRUE,labels=2,lines=
0) cl$cluster cl$centers
"agglomarative clustering " clusters<-hclust(dist(iris[,3:4]))
plot(clusters) clusterCut<-cutree(clusters,3)
table(clusterCut,iris$Species)
```

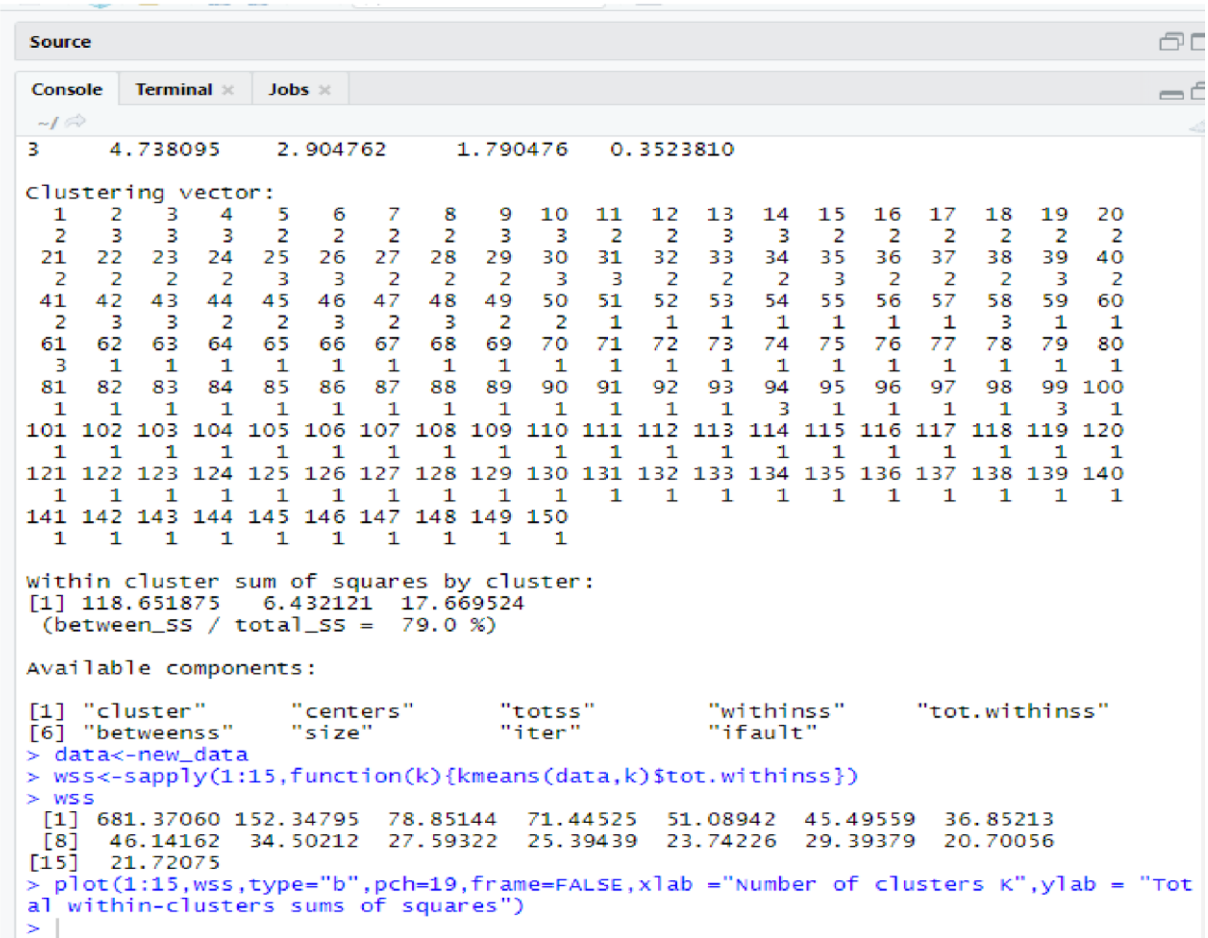
Output :



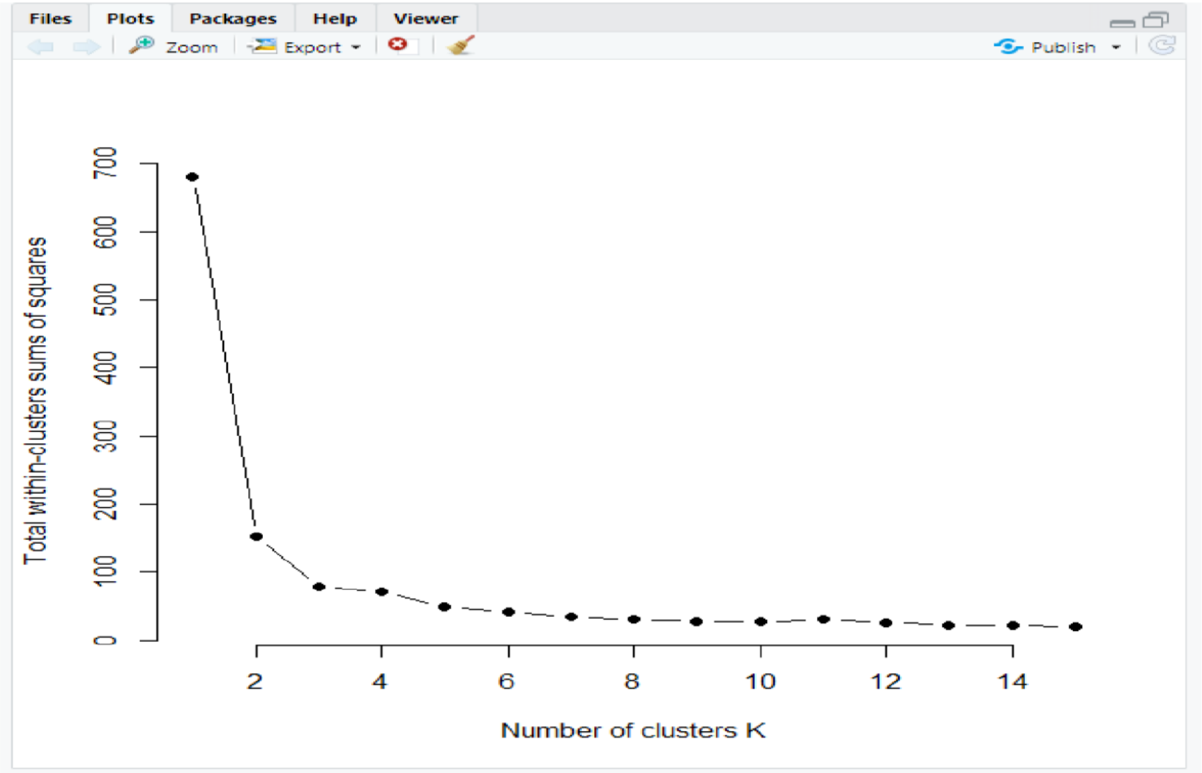
```
> "K-means Clustering "
[1] "K-means Clustering "
> data(iris)
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> new_data<-subset(iris,select = c(-Species))
> new_data
    Sepal.Length Sepal.Width Petal.Length Petal.Width
1            5.1         3.5          1.4         0.2
2            4.9         3.0          1.4         0.2
3            4.7         3.2          1.3         0.2
4            4.6         3.1          1.5         0.2
5            5.0         3.6          1.4         0.2
6            5.4         3.9          1.7         0.4
7            4.6         3.4          1.4         0.3
8            5.0         3.4          1.5         0.2
9            4.4         2.9          1.4         0.2
10           4.9         3.1          1.5         0.1
11           5.4         3.7          1.5         0.2
12           4.8         3.4          1.6         0.2
13           4.8         3.0          1.4         0.1
14           4.3         3.0          1.1         0.1
15           5.8         4.0          1.2         0.2
16           5.7         4.4          1.5         0.4
17           5.4         3.9          1.3         0.4
18           5.1         3.5          1.4         0.3
19           5.7         3.8          1.7         0.3
20           5.1         3.8          1.5         0.3
21           5.4         3.4          1.7         0.2
22           5.1         3.7          1.5         0.4
23           4.6         3.6          1.0         0.2
24           5.1         3.3          1.7         0.5
25           4.8         3.4          1.9         0.2
26           5.0         3.0          1.6         0.2
27           5.0         3.4          1.6         0.4
28           5.2         3.5          1.5         0.2
29           5.2         3.4          1.4         0.2
30           4.7         3.2          1.6         0.2
```

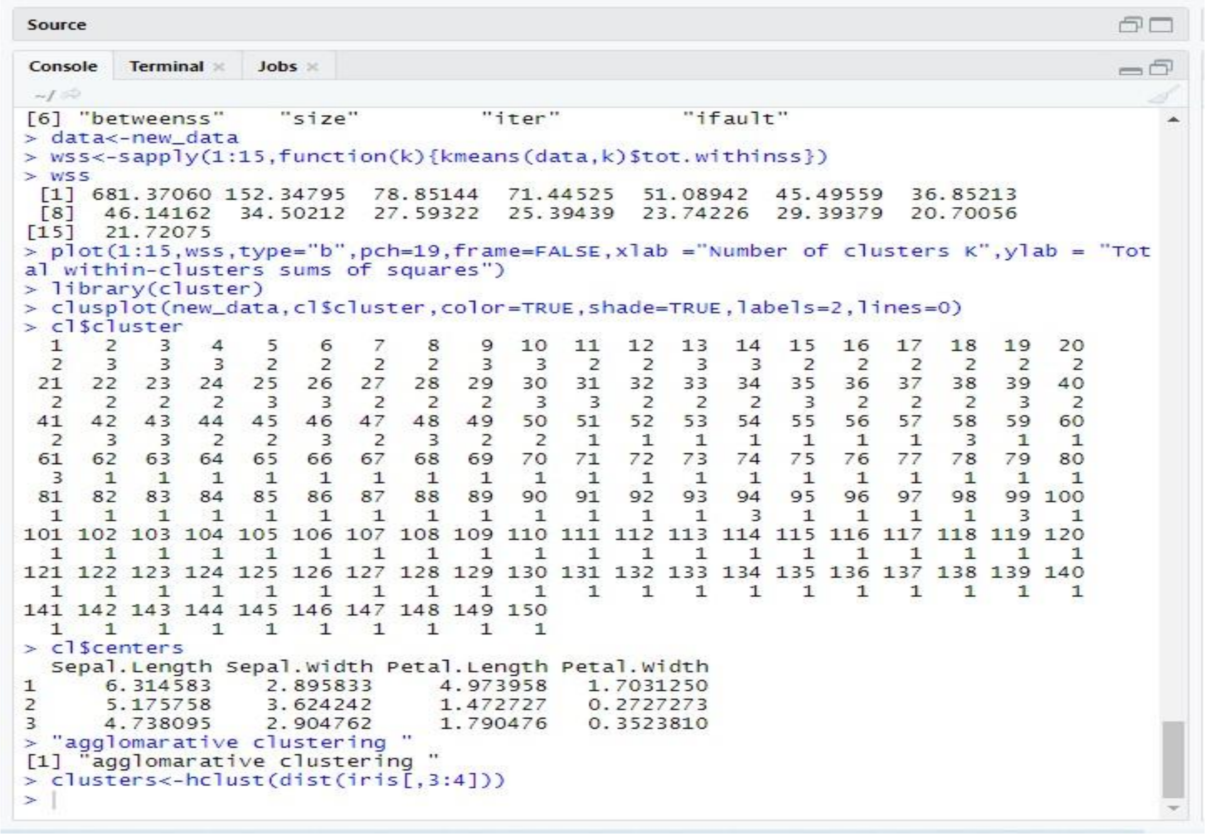


```
> cl<-kmeans(new_data,3)
> cl
K-means clustering with 3 clusters of sizes 96, 33, 21

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.314583    2.895833     4.973958   1.7031250
2     5.175758    3.624242     1.472727   0.2727273
3     4.738095    2.904762     1.790476   0.3523810

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
  2   3   3   3   2   2   2   2   3   3   2   2   3   3   2   2   2   2   2   2 
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 
  2   2   2   2   3   3   2   2   2   3   3   2   2   2   3   2   2   2   3   2 
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 
  2   3   3   2   2   3   2   3   2   2   1   1   1   1   1   1   1   3   1   1 
```

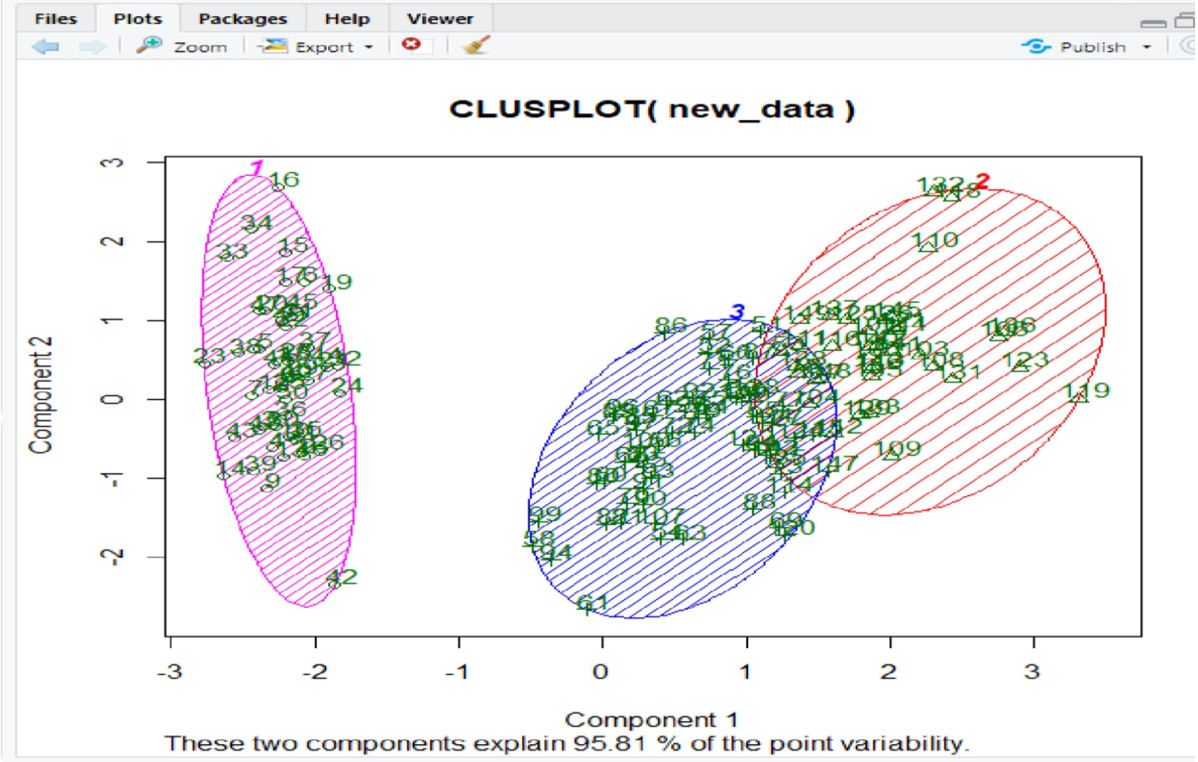


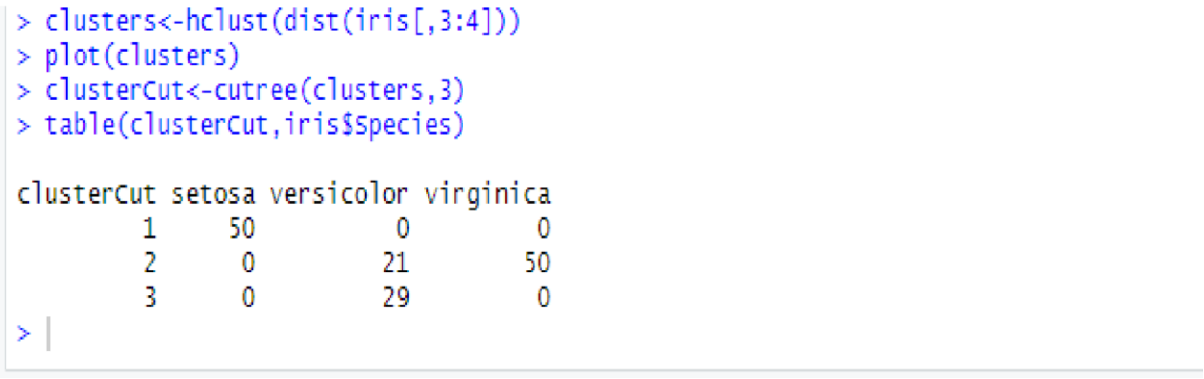
```
3      4.738095     2.904762      1.790476    0.3523810

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
  2   3   3   3   2   2   2   2   3   3   2   2   3   3   2   2   2   2   2   2 
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 
  2   2   2   2   3   3   2   2   2   3   3   2   2   2   3   2   2   2   3   2 
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 
  2   3   3   2   2   3   2   3   2   2   1   1   1   1   1   1   1   3   1   1 
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 
  3   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
 81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 
  1   1   1   1   1   1   1   1   1   1   1   1   1   3   1   1   1   1   3   1 
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
141 142 143 144 145 146 147 148 149 150 
  1   1   1   1   1   1   1   1   1   1 

Within cluster sum of squares by cluster:
[1] 118.651875   6.432121  17.669524
 (between_SS / total_SS =  79.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"      
> data<-new_data
> wss<-sapply(1:15,function(k){kmeans(data,k)$tot.withinss})
> wss
 [1] 681.37060 152.34795  78.85144  71.44525  51.08942  45.49559  36.85213
 [8]  46.14162  34.50212  27.59322  25.39439  23.74226  29.39379  20.70056
[15]  21.72075
> plot(1:15,wss,type="b",pch=19,frame=FALSE,xlab ="Number of clusters K",ylab = "Tot
al within-clusters sums of squares")
> |
```



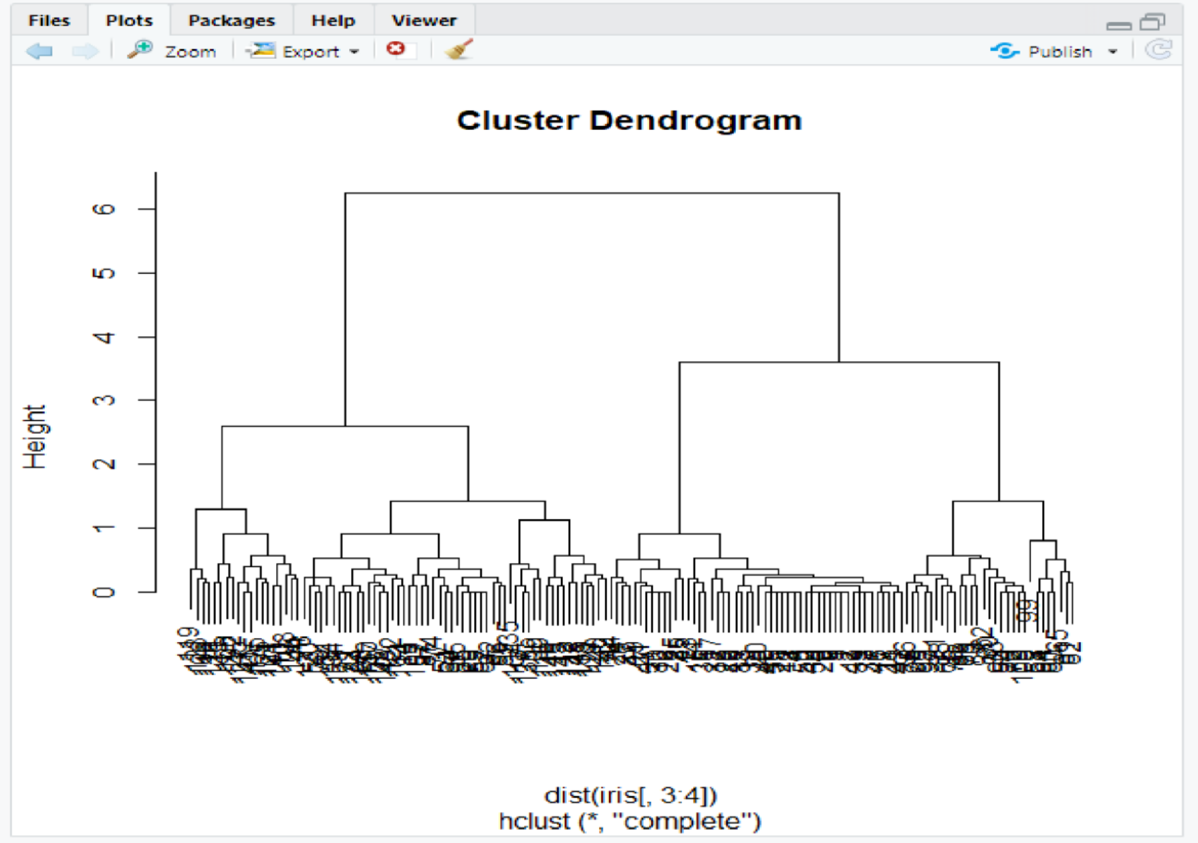


```
[6] "betweenss"    "size"         "iter"         "ifault"
> data<-new_data
> wss<-sapply(1:15,function(k){kmeans(data,k)$tot.withinss})
> wss
 [1] 681.37060 152.34795  78.85144  71.44525  51.08942  45.49559  36.85213
 [8]  46.14162  34.50212  27.59322  25.39439  23.74226  29.39379  20.70056
[15]  21.72075
> plot(1:15,wss,type="b",pch=19,frame=FALSE,xlab ="Number of clusters K",ylab = "Tot
al within-clusters sums of squares")
> library(cluster)
> clusplot(new_data,cl$cluster,color=TRUE,shade=TRUE,labels=2,lines=0)
> cl$cluster
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
  2   3   3   3   2   2   2   2   3   3   2   2   3   3   2   2   2   2   2   2 
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 
  2   2   2   2   3   3   2   2   2   3   3   2   2   2   3   2   2   2   3   2 
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 
  2   3   3   2   2   3   2   3   2   2   1   1   1   1   1   1   1   3   1   1 
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 
  3   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
 81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 
  1   1   1   1   1   1   1   1   1   1   1   1   1   3   1   1   1   1   3   1 
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
141 142 143 144 145 146 147 148 149 150 
  1   1   1   1   1   1   1   1   1   1 
> cl$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.314583    2.895833     4.973958   1.7031250
2     5.175758    3.624242     1.472727   0.2727273
3     4.738095    2.904762     1.790476   0.3523810
> "agglomarative clustering "
[1] "agglomarative clustering "
> clusters<-hclust(dist(iris[,3:4]))
> |
```





```
> clusters<-hclust(dist(iris[,3:4]))
> plot(clusters)
> clusterCut<-cutree(clusters,3)
> table(clusterCut,iris$Species)

clusterCut setosa versicolor virginica
         1     50          0         0
         2      0         21        50
         3      0         29         0
> 
```



Practical-5

Aim: Practical of Time-series forecasting

#consider the inbuilt data set Air Passengers data("AirPassengers")

class(AirPassengers) start(AirPassengers) #to know the end of time

series end(AirPassengers) frequency(AirPassengers)

#to know the mean median etc of the dataset summary(AirPassengers)

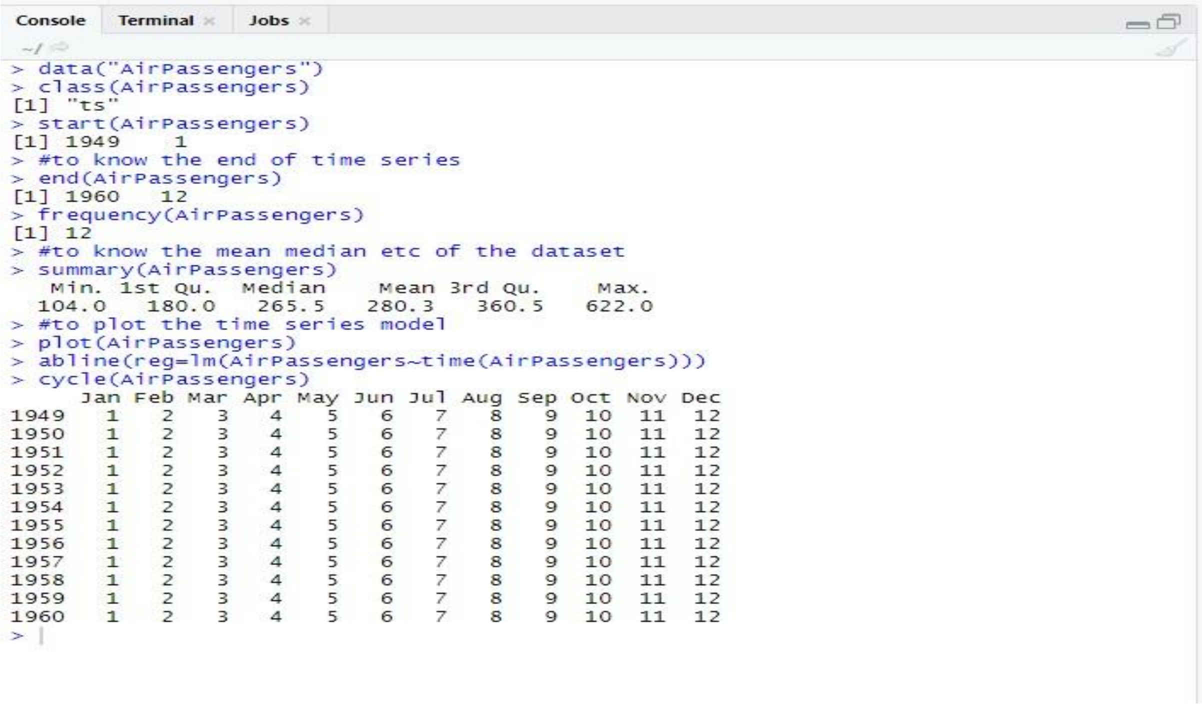
#to plot the time series model plot(AirPassengers)

abline(reg=lm(AirPassengers~time(AirPassengers))) cycle(AirPassengers)

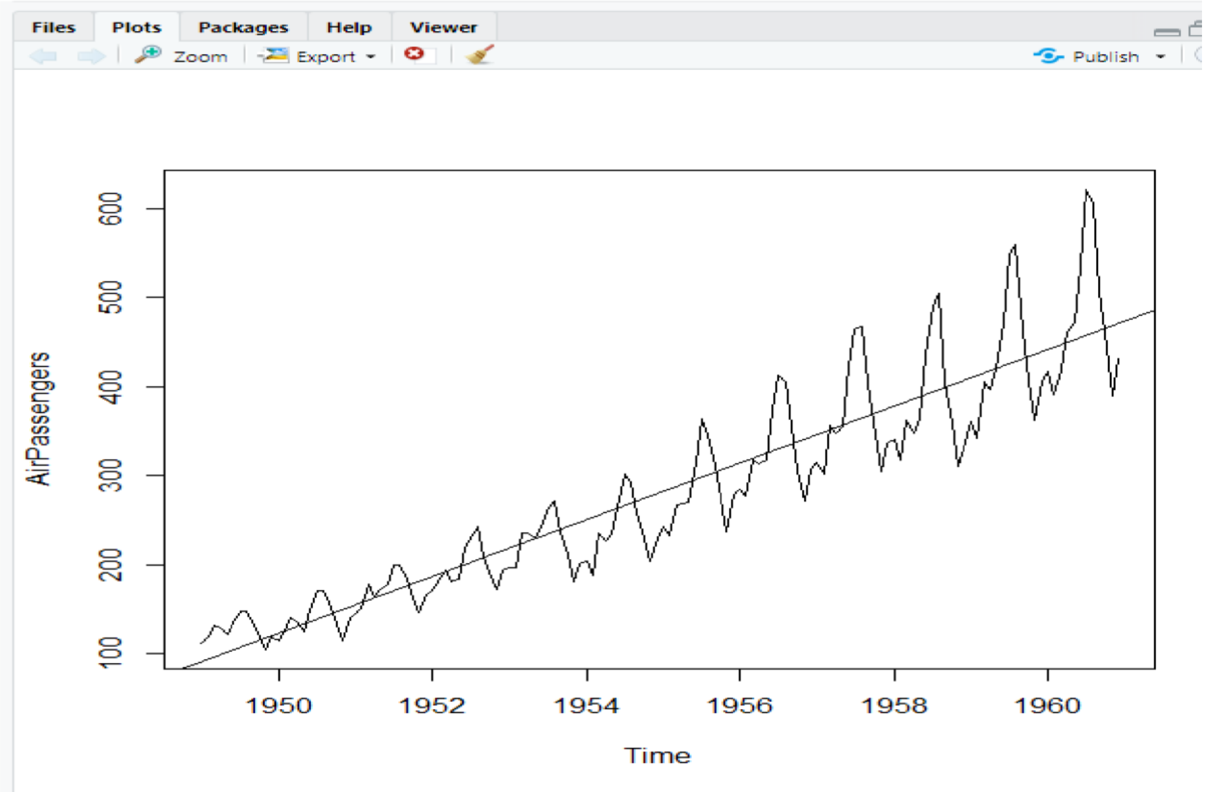
plot(aggregate(AirPassengers,FUN=mean))

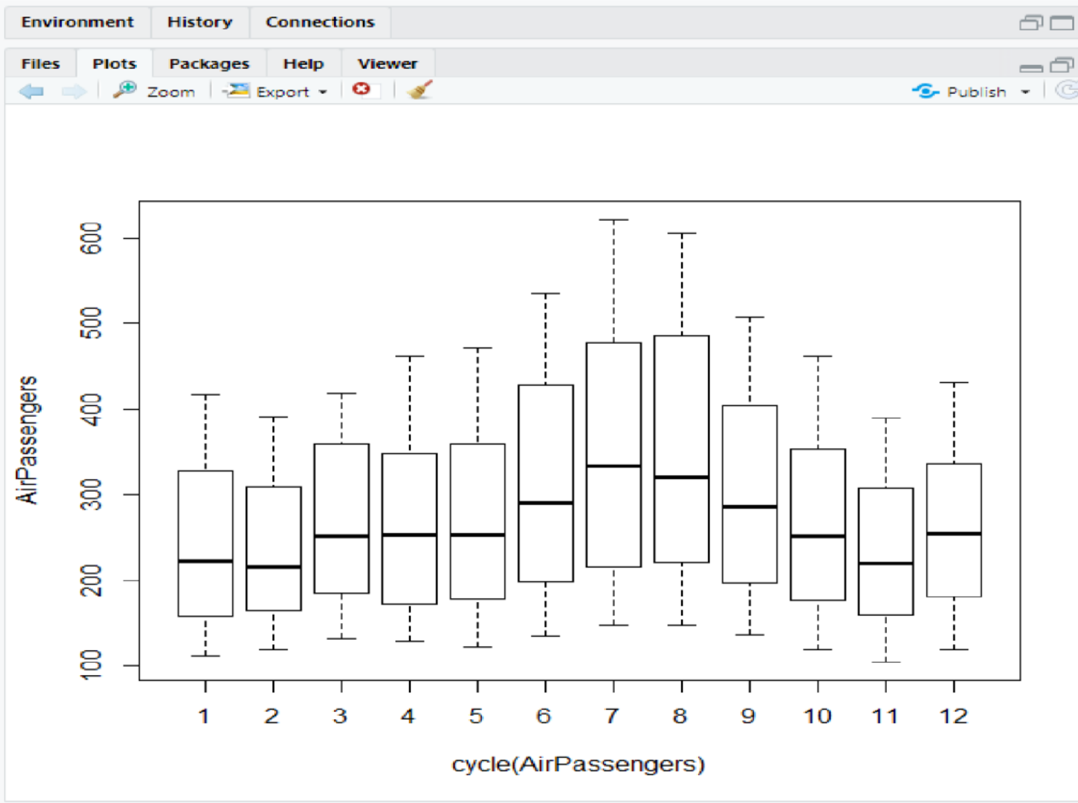
boxplot(AirPassengers~cycle(AirPassengers))

Output :



```
> data("AirPassengers")
> class(AirPassengers)
[1] "ts"
> start(AirPassengers)
[1] 1949    1
> #to know the end of time series
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> #to know the mean median etc of the dataset
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  104.0   180.0   265.5   280.3   360.5   622.0 
> #to plot the time series model
> plot(AirPassengers)
> abline(reg=lm(AirPassengers~time(AirPassengers)))
> cycle(AirPassengers)
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949   1   2   3   4   5   6   7   8   9  10  11  12
1950   1   2   3   4   5   6   7   8   9  10  11  12
1951   1   2   3   4   5   6   7   8   9  10  11  12
1952   1   2   3   4   5   6   7   8   9  10  11  12
1953   1   2   3   4   5   6   7   8   9  10  11  12
1954   1   2   3   4   5   6   7   8   9  10  11  12
1955   1   2   3   4   5   6   7   8   9  10  11  12
1956   1   2   3   4   5   6   7   8   9  10  11  12
1957   1   2   3   4   5   6   7   8   9  10  11  12
1958   1   2   3   4   5   6   7   8   9  10  11  12
1959   1   2   3   4   5   6   7   8   9  10  11  12
1960   1   2   3   4   5   6   7   8   9  10  11  12
> 
```




Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
AirPassengers
100
200
300
400
500
600
1
2
3
4
5
6
7
8
9
10
11
12
cycle(AirPassengers)

Practical-6

Aim: Practical of Simple/Multiple Linear Regression

#consider some data set height<-

c(102,117,105,141,135,115,138,144,137,100,131,119,115,121,113) weight<-

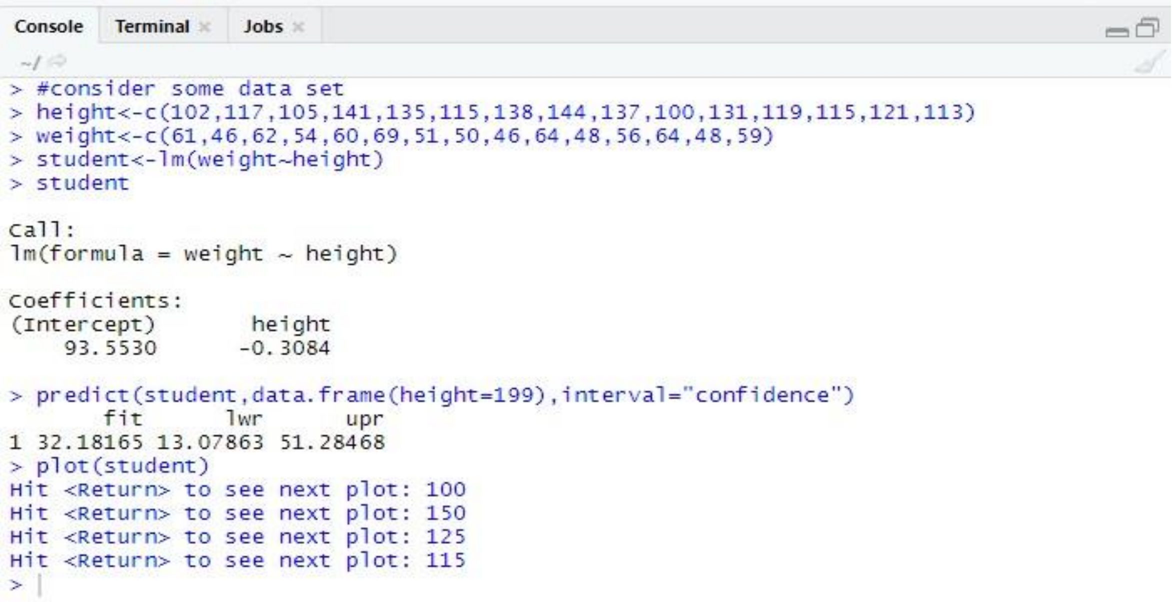
c(61,46,62,54,60,69,51,50,46,64,48,56,64,48,59)

#lm is for Linear Regression student<-lm(weight~height) student

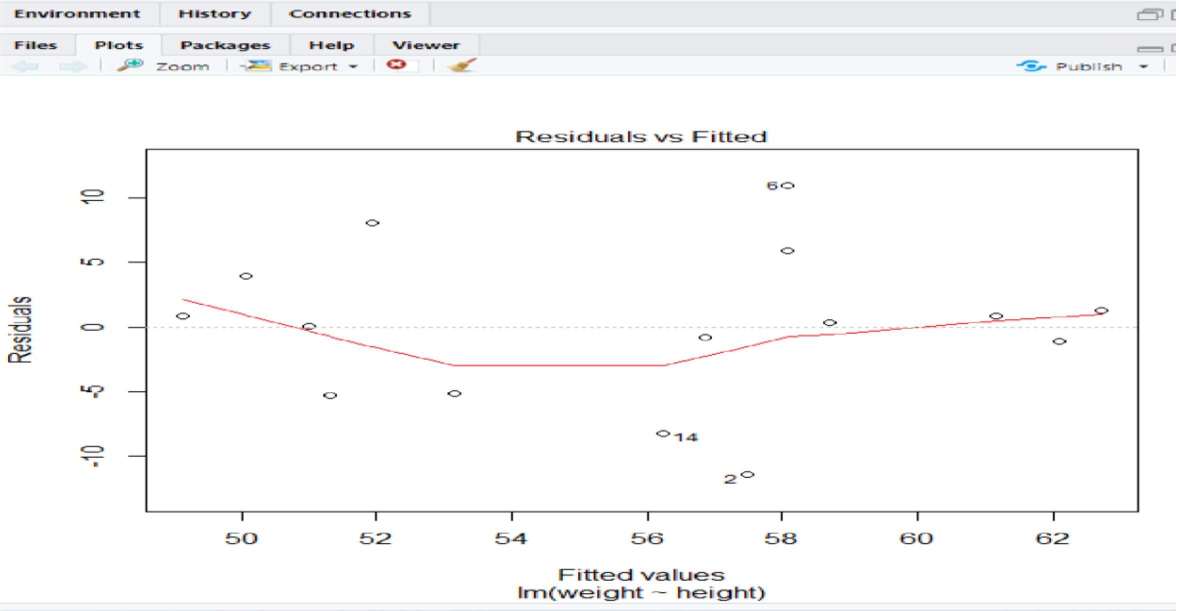
#to predict use predict command predict(student,data.frame(height=199),interval="confidence")

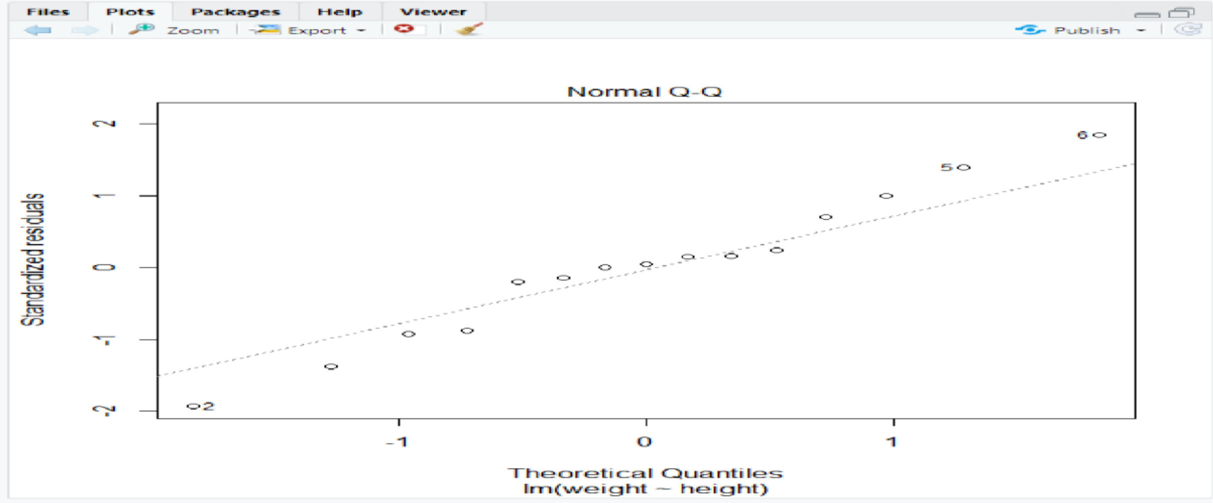
#to plot the data plot(student)

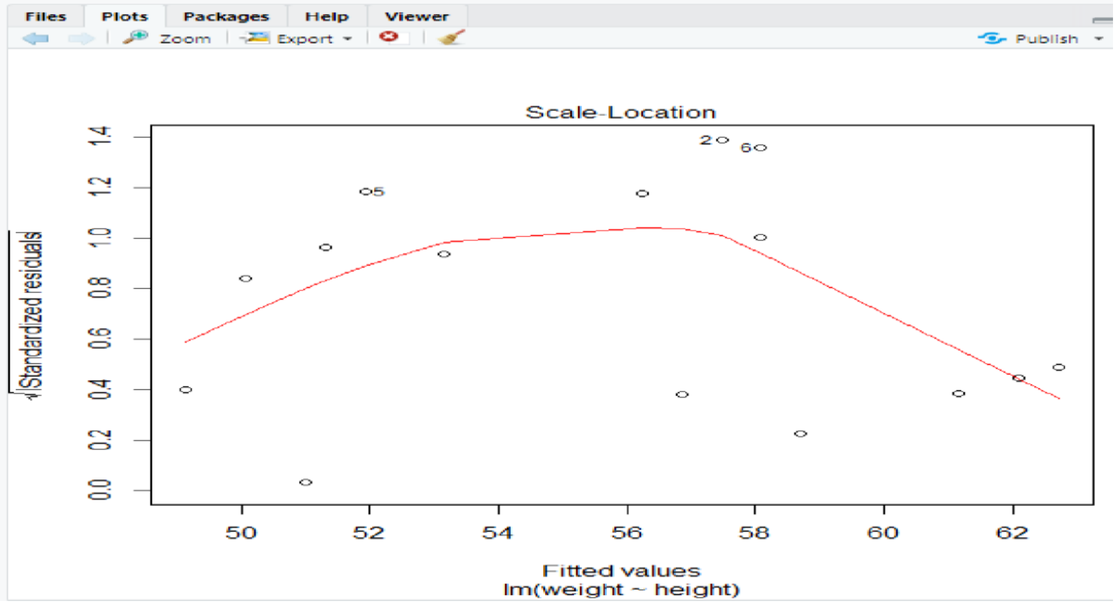
Output :

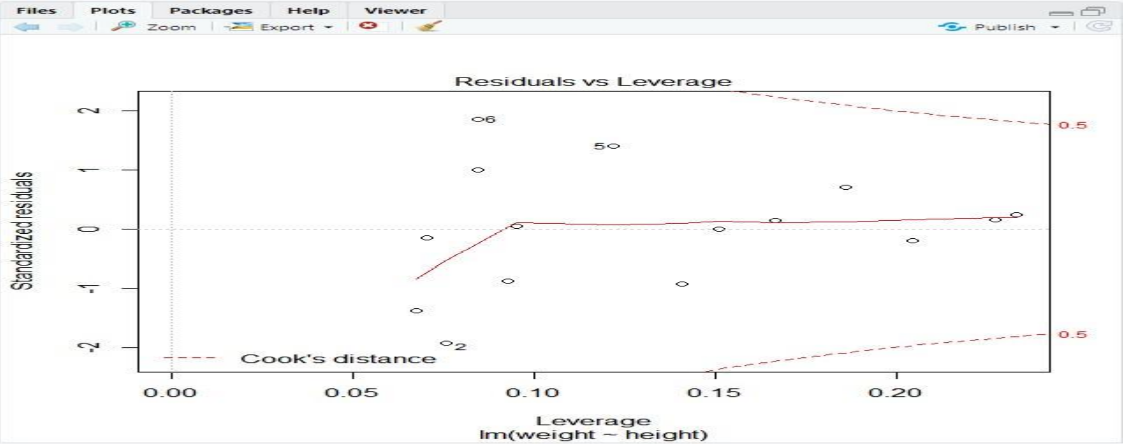


```
> #consider some data set
> height<-c(102,117,105,141,135,115,138,144,137,100,131,119,115,121,113)
> weight<-c(61,46,62,54,60,69,51,50,46,64,48,56,64,48,59)
> student<-lm(weight~height)
> student

Call:
lm(formula = weight ~ height)

Coefficients:
(Intercept)       height
    93.5530      -0.3084

> predict(student,data.frame(height=199),interval="confidence")
       fit      lwr      upr
1 32.18165 13.07863 51.28468
> plot(student)
Hit <Return> to see next plot: 100
Hit <Return> to see next plot: 150
Hit <Return> to see next plot: 125
Hit <Return> to see next plot: 115
> 
```




Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Normal Q-Q
Standardized residuals
-2
-1
0
1
2
-1
0
1
6
5
2
Theoretical Quantiles
lm(weight ~ height)


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Scale-Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
50
52
54
56
58
60
62
2
6
5
Fitted values
lm(weight ~ height)

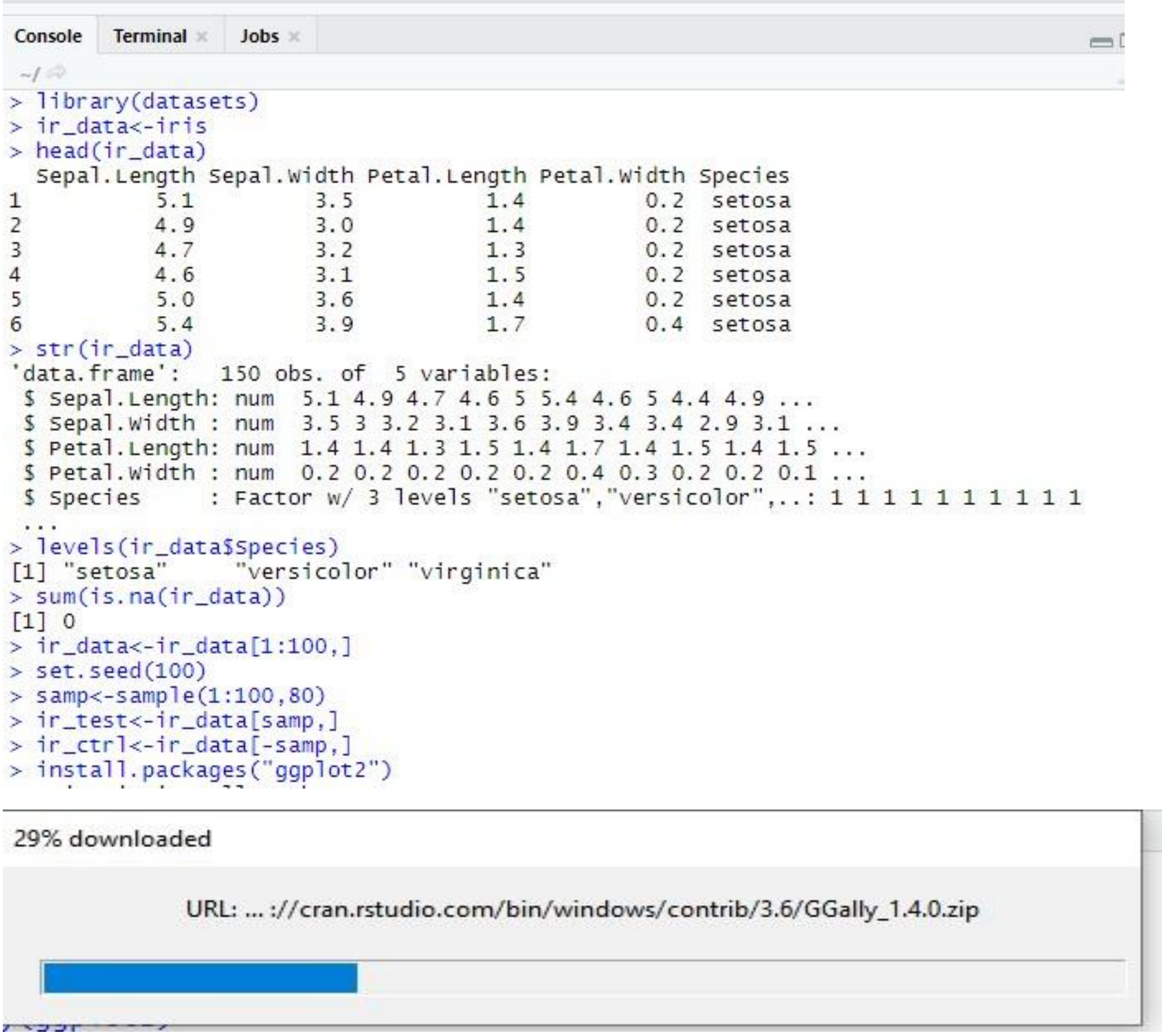

Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
0.00
0.05
0.10
0.15
0.20
6
5
2
0.5
0.5
Cook's distance
Leverage
lm(weight ~ height)

Practical-7

Aim: Practical of Logistics Regression

```
library(datasets) ir_data<-iris head(ir_data) str(ir_data) levels(ir_data$Species)
sum(is.na(ir_data)) ir_data<-ir_data[1:100,] set.seed(100) samp<-
sample(1:100,80) ir_test<-ir_data[samp,] ir_ctrl<-ir_data[-samp,]
install.packages("ggplot2") library(ggplot2) library(ggplot2)
install.packages("GGally") ggpairs(ir_test) y<-ir_test$Species; x<-
ir_test$Sepal.Length glfit<-glm(y~x,family='binomial') summary(glfit) newdata<-
data.frame(x=ir_ctrl$Sepal.Length) predicted_val<-
predict(glfit,newdata,type="response") prediction<-
data.frame(ir_ctrl$Sepal.Length,ir_ctrl$Species,predicted_val) prediction
qplot(prediction[,1],round(prediction[,3]),col=prediction[,2],xlab='sepal. Length',ylab='prediction using
logistic Reg')
```

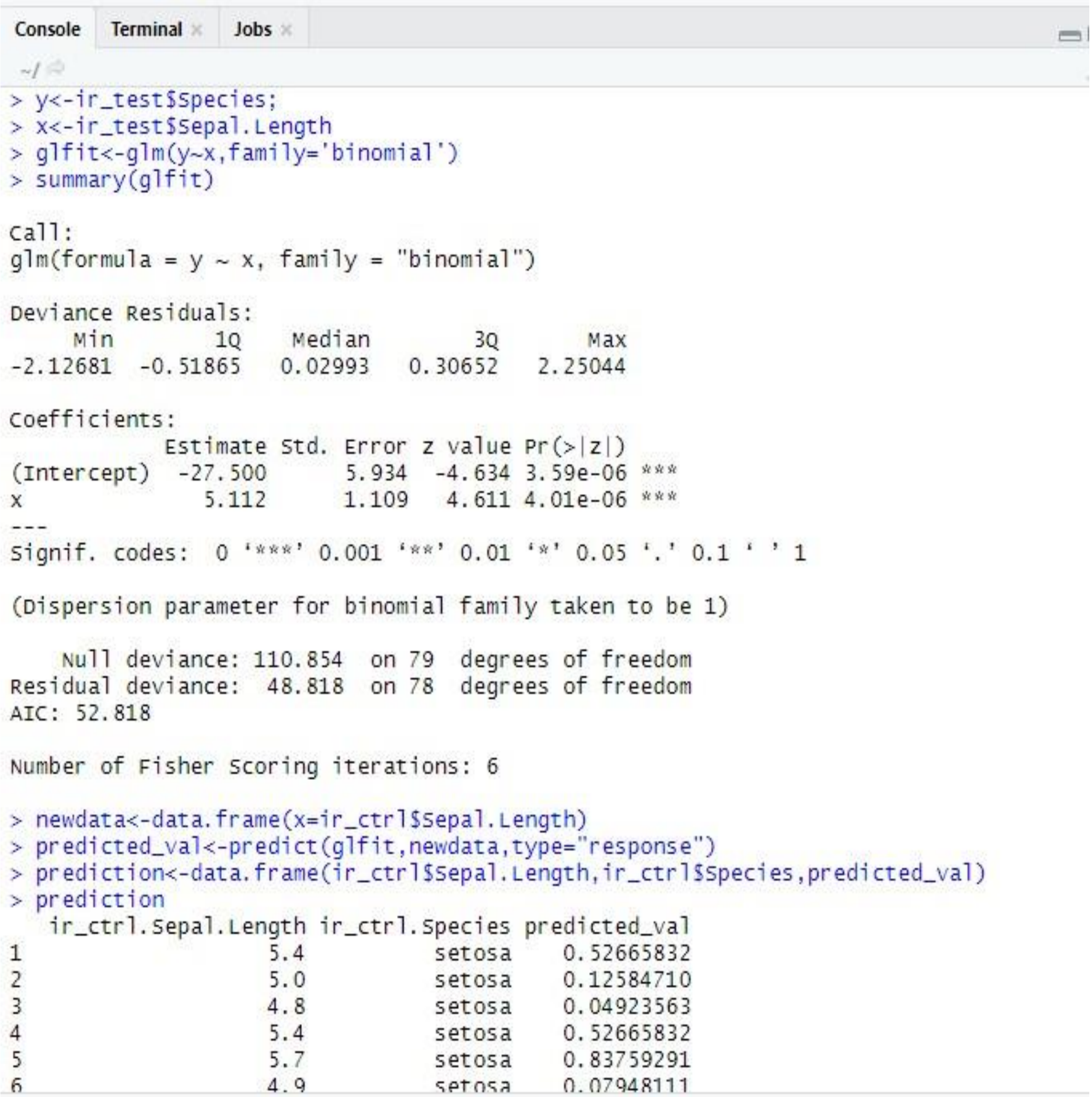
Output :



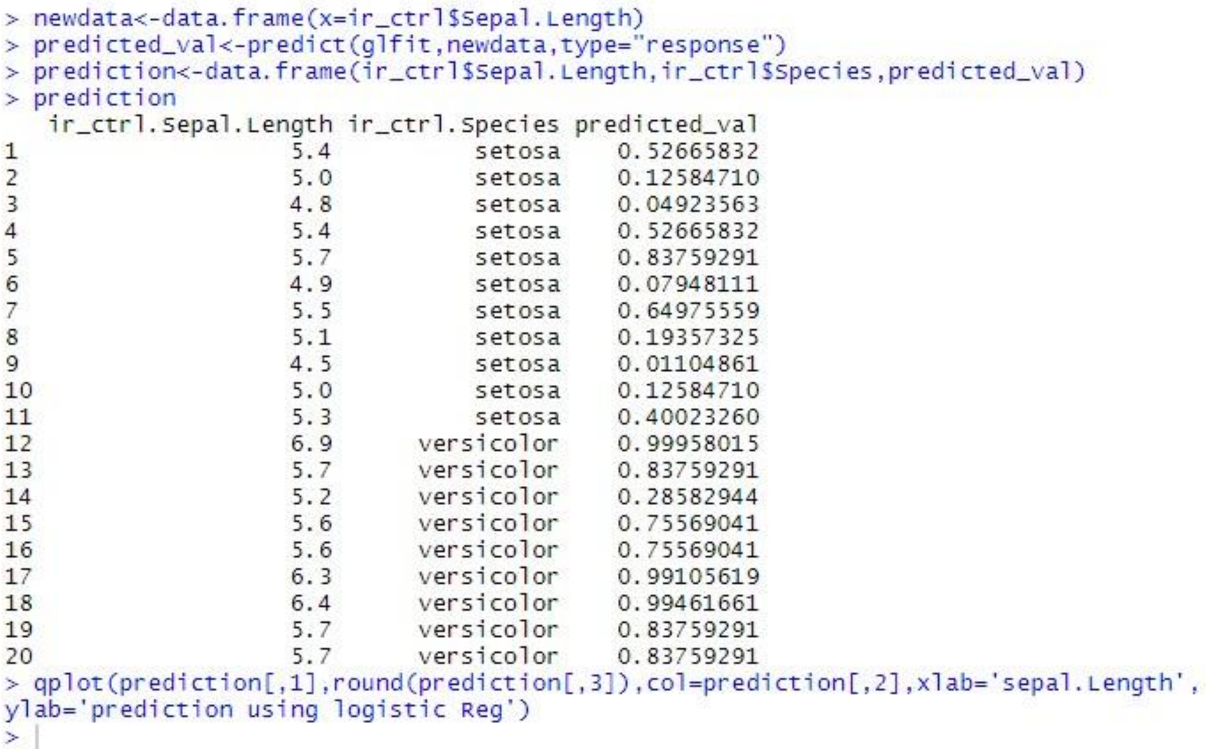
```
> library(datasets)
> ir_data<-iris
> head(ir_data)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> str(ir_data)
'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 1
 ...
> levels(ir_data$Species)
[1] "setosa"     "versicolor" "virginica"
> sum(is.na(ir_data))
[1] 0
> ir_data<-ir_data[1:100,]
> set.seed(100)
> samp<-sample(1:100,80)
> ir_test<-ir_data[samp,]
> ir_ctrl<-ir_data[-samp,]
> install.packages("ggplot2")
```

29% downloaded

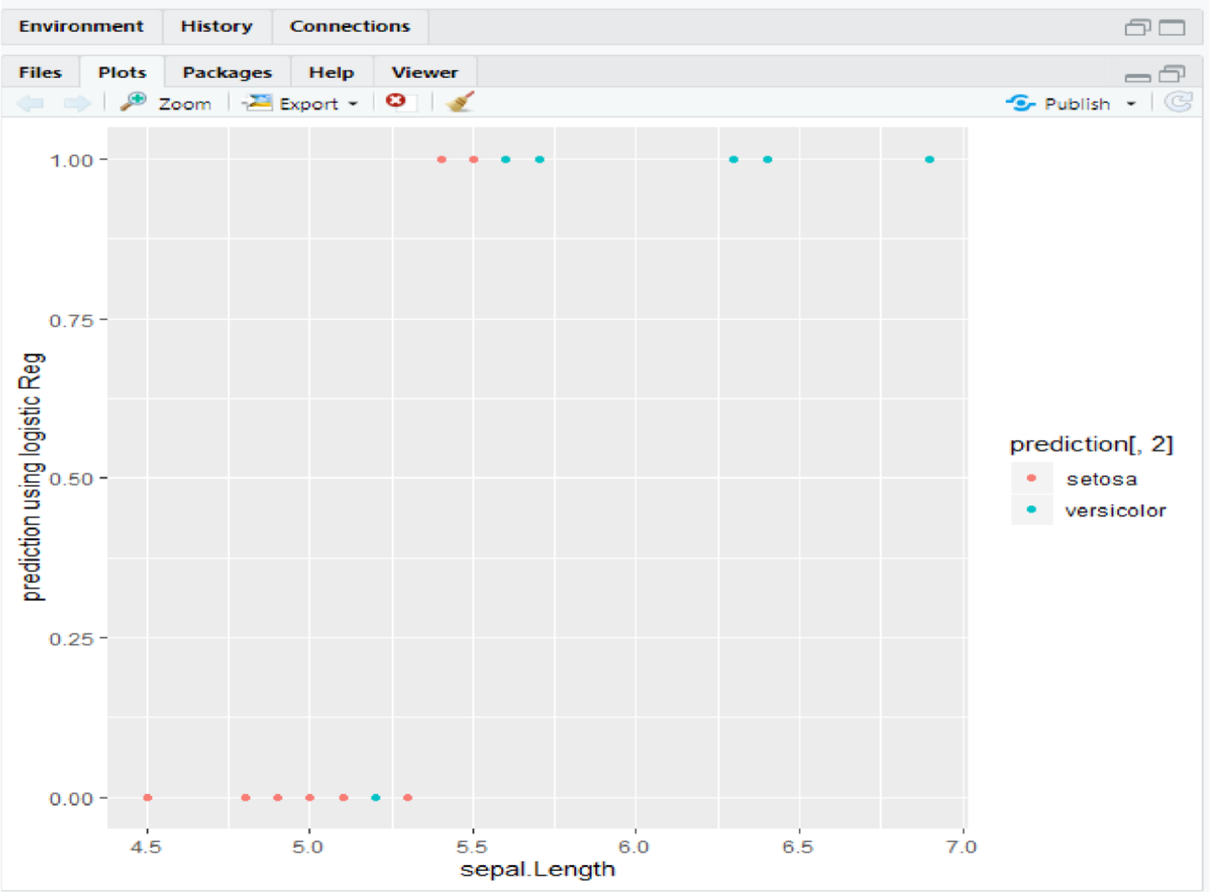
URL: ... ://cran.rstudio.com/bin/windows/contrib/3.6/GGally_1.4.0.zip



```
> y<-ir_test$Species;
> x<-ir_test$Sepal.Length
> glfit<-glm(y~x,family='binomial')
> summary(glfit)

Call:
glm(formula = y ~ x, family = "binomial")

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.12681  -0.51865   0.02993   0.30652   2.25044

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -27.500      5.934  -4.634 3.59e-06 ***
x              5.112      1.109   4.611 4.01e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 110.854  on 79  degrees of freedom
Residual deviance:  48.818  on 78  degrees of freedom
AIC: 52.818

Number of Fisher Scoring iterations: 6

> newdata<-data.frame(x=ir_ctrl$Sepal.Length)
> predicted_val<-predict(glfit,newdata,type="response")
> prediction<-data.frame(ir_ctrl$Sepal.Length,ir_ctrl$Species,predicted_val)
> prediction
   ir_ctrl.Sepal.Length ir_ctrl.Species predicted_val
1                   5.4          setosa    0.52665832
2                   5.0          setosa    0.12584710
3                   4.8          setosa    0.04923563
4                   5.4          setosa    0.52665832
5                   5.7          setosa    0.83759291
6                   4.9          setosa    0.07948111
```



```
> newdata<-data.frame(x=ir_ctrl$Sepal.Length)
> predicted_val<-predict(glfit,newdata,type="response")
> prediction<-data.frame(ir_ctrl$Sepal.Length,ir_ctrl$Species,predicted_val)
> prediction
   ir_ctrl.Sepal.Length ir_ctrl.Species predicted_val
1                   5.4          setosa    0.52665832
2                   5.0          setosa    0.12584710
3                   4.8          setosa    0.04923563
4                   5.4          setosa    0.52665832
5                   5.7          setosa    0.83759291
6                   4.9          setosa    0.07948111
7                   5.5          setosa    0.64975559
8                   5.1          setosa    0.19357325
9                   4.5          setosa    0.01104861
10                  5.0          setosa    0.12584710
11                  5.3          setosa    0.40023260
12                  6.9      versicolor    0.99958015
13                  5.7      versicolor    0.83759291
14                  5.2      versicolor    0.28582944
15                  5.6      versicolor    0.75569041
16                  5.6      versicolor    0.75569041
17                  6.3      versicolor    0.99105619
18                  6.4      versicolor    0.99461661
19                  5.7      versicolor    0.83759291
20                  5.7      versicolor    0.83759291
> qplot(prediction[,1],round(prediction[,3]),col=prediction[,2],xlab='sepal.Length',
ylab='prediction using logistic Reg')
> 
```



Practical-8

Aim: Practical of Hypothesis testing

#Entering the data

>x=c(6.2,6.6,7.1,7.4,7.6,7.9,8,8.3,8.4,8.5,8.6,8.8,8.8,9.1,9.2,9.4,9.7,9. 9,10.2,10.4,10.8,11.3,11.9)

#one-sample Hypothesis test

>t.test(x-9,alternative = "two.sided",conf.level = 0.95)

#Two-sample hypothesis test

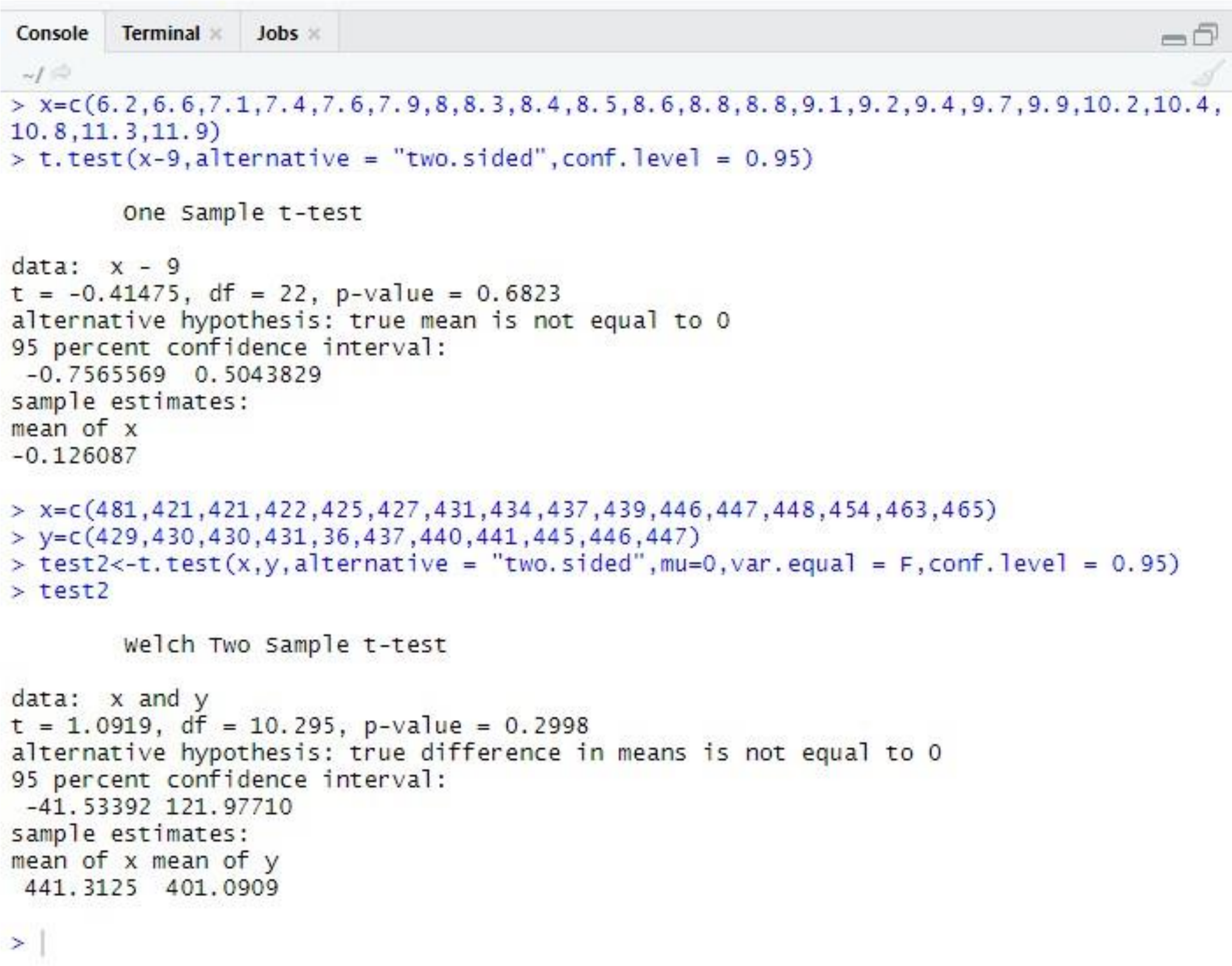
>x=c(481,421,421,422,425,427,431,434,437,439,446,447,448,454,46 3,465)

>y=c(429,430,430,431,36,437,440,441,445,446,447)

>test2<-t.test(x,y,alternative = "two.sided",mu=0,var.equal = F,conf.level = 0.95)

>test2

Output :



```
> x=c(6.2,6.6,7.1,7.4,7.6,7.9,8,8.3,8.4,8.5,8.6,8.8,8.8,9.1,9.2,9.4,9.7,9.9,10.2,10.4,
10.8,11.3,11.9)
> t.test(x-9,alternative = "two.sided",conf.level = 0.95)

        One Sample t-test

data:  x - 9
t = -0.41475, df = 22, p-value = 0.6823
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.7565569  0.5043829
sample estimates:
mean of x
-0.126087

> x=c(481,421,421,422,425,427,431,434,437,439,446,447,448,454,463,465)
> y=c(429,430,430,431,36,437,440,441,445,446,447)
> test2<-t.test(x,y,alternative = "two.sided",mu=0,var.equal = F,conf.level = 0.95)
> test2

        Welch Two Sample t-test

data:  x and y
t = 1.0919, df = 10.295, p-value = 0.2998
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -41.53392 121.97710
sample estimates:
mean of x mean of y
 441.3125  401.0909

> |
```

✸ Interpretation of the result:

The p-value (0.2998) is greater than the significance level 5%(1-0.95)

So,we conclude that the null hypothesis that the population means are equal is plausible.

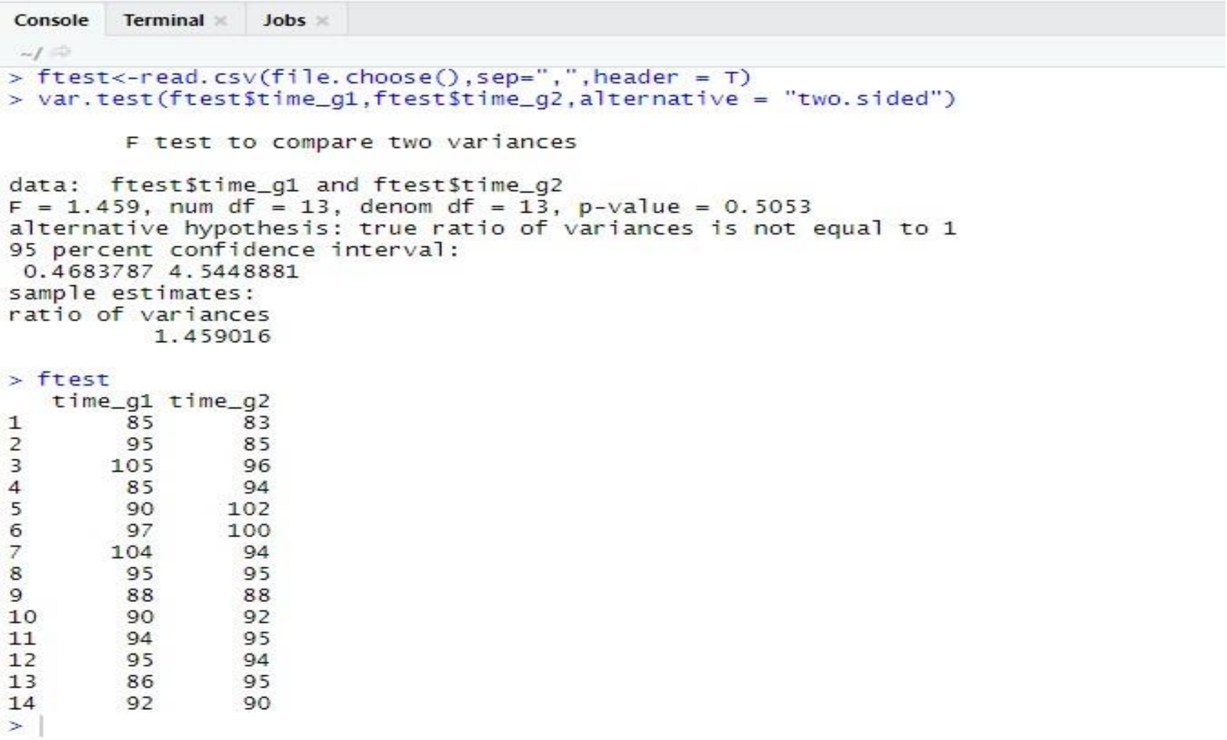
Practical-9

Aim: Practical of Analysis of Variance

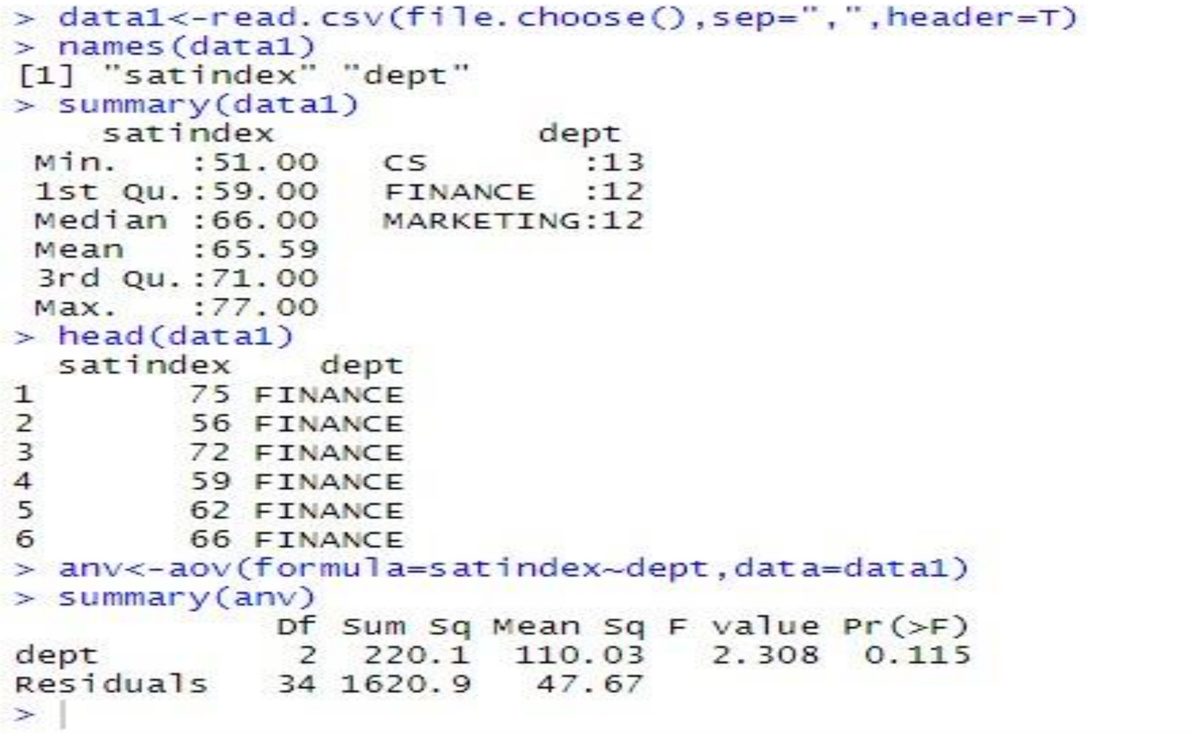
ftest<-read.csv(file.choose(),sep=",",header = T) var.test(ftest$time_g1,ftest$time_g2,alternative = "two.sided")

//one way anova names(data1) summary(data1) head(data1) anv<-aov(formula=satindex~dept,data=data1) summary(anv) //two way anova data2<-read.csv(file.choose(),sep=",",header=T) names(data2) summary(data2) head(data2) anv1<-aov(formula = satindex~dept+exp+dept*exp,data=data2) summary(anv1)

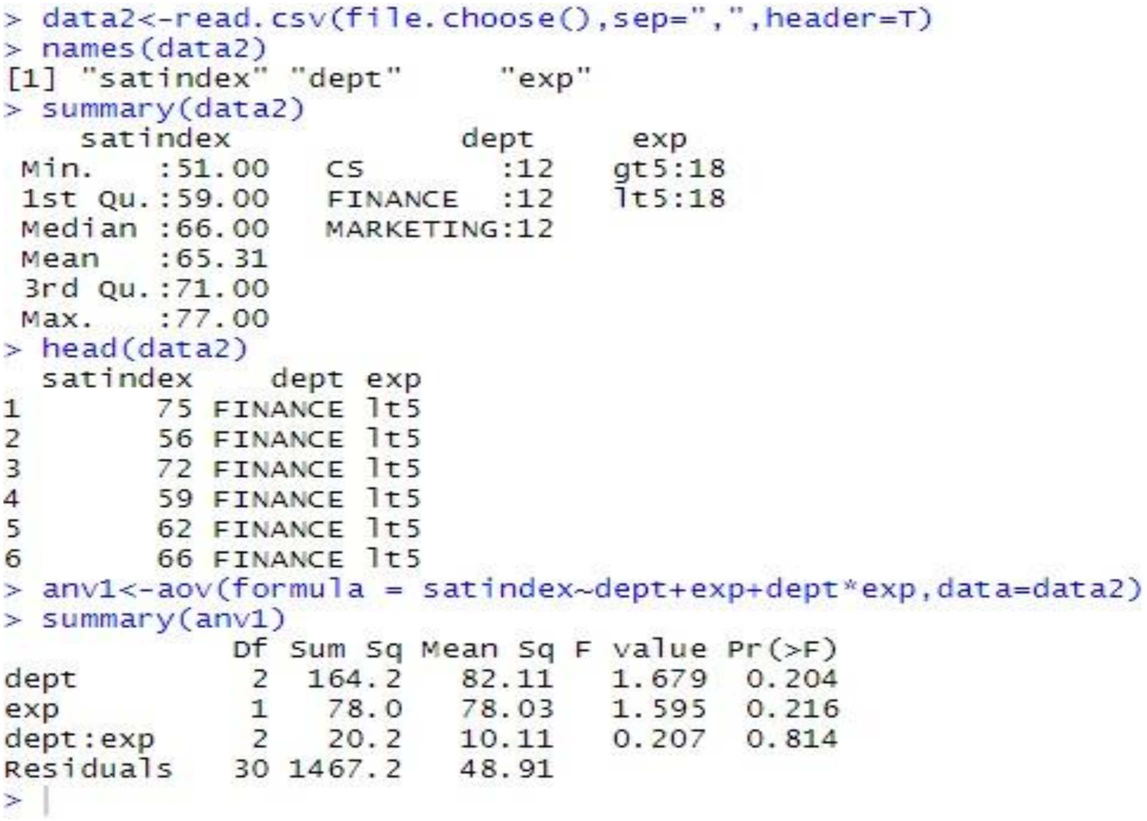
Output :



```
> ftest<-read.csv(file.choose(),sep=",",header = T)
> var.test(ftest$time_g1,ftest$time_g2,alternative = "two.sided")

        F test to compare two variances

data:  ftest$time_g1 and ftest$time_g2
F = 1.459, num df = 13, denom df = 13, p-value = 0.5053
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4683787 4.5448881
sample estimates:
ratio of variances
          1.459016

> ftest
   time_g1 time_g2
1       85      83
2       95      85
3      105      96
4       85      94
5       90     102
6       97     100
7      104      94
8       95      95
9       88      88
10      90      92
11      94      95
12      95      94
13      86      95
14      92      90
> |
```



```
> data1<-read.csv(file.choose(),sep=",",header=T)
> names(data1)
[1] "satindex" "dept"
> summary(data1)
    satindex             dept   
 Min.   :51.00   CS         :13  
 1st Qu.:59.00   FINANCE    :12  
 Median :66.00   MARKETING:12  
 Mean   :65.59                  
 3rd Qu.:71.00                  
 Max.   :77.00                  
> head(data1)
  satindex    dept
1       75 FINANCE
2       56 FINANCE
3       72 FINANCE
4       59 FINANCE
5       62 FINANCE
6       66 FINANCE
> anv<-aov(formula=satindex~dept,data=data1)
> summary(anv)
            Df Sum Sq Mean Sq F value Pr(>F)
dept         2  220.1  110.03   2.308  0.115
Residuals   34 1620.9   47.67               
> 
```



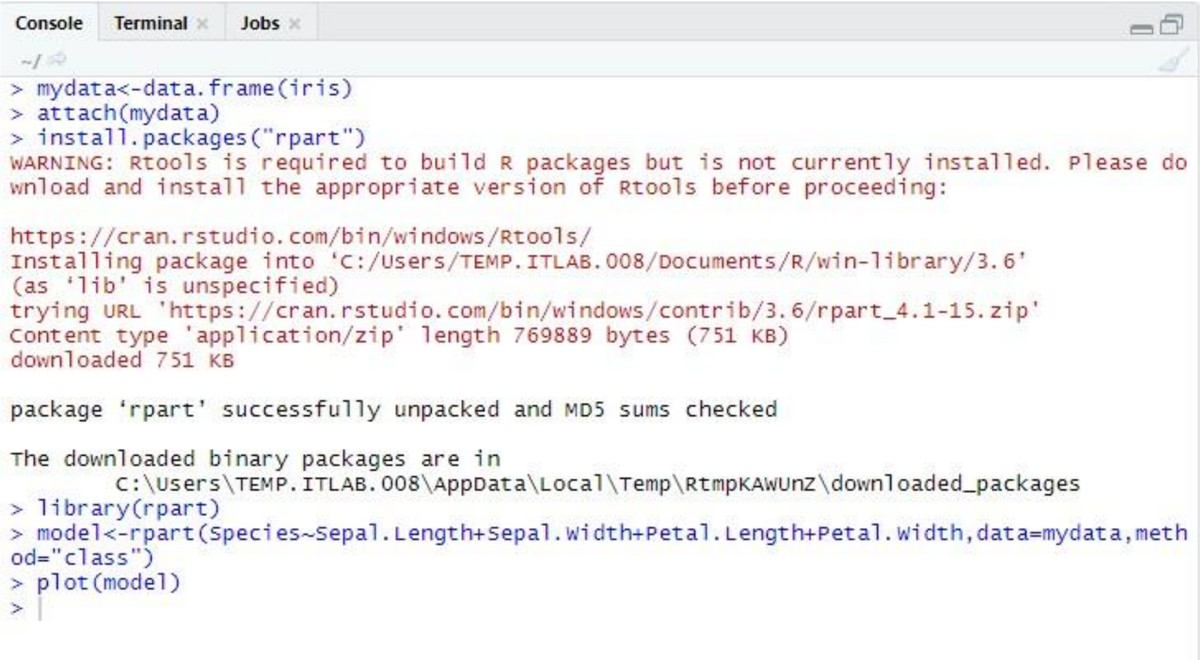
```
> data2<-read.csv(file.choose(),sep=",",header=T)
> names(data2)
[1] "satindex" "dept"     "exp"
> summary(data2)
    satindex            dept      exp   
 Min.   :51.00   CS         :12   gt5:18  
 1st Qu.:59.00   FINANCE    :12   lt5:18  
 Median :66.00   MARKETING:12           
 Mean   :65.31                          
 3rd Qu.:71.00                          
 Max.   :77.00                          
> head(data2)
  satindex    dept exp
1       75 FINANCE lt5
2       56 FINANCE lt5
3       72 FINANCE lt5
4       59 FINANCE lt5
5       62 FINANCE lt5
6       66 FINANCE lt5
> anv1<-aov(formula = satindex~dept+exp+dept*exp,data=data2)
> summary(anv1)
            Df Sum Sq Mean Sq F value Pr(>F)
dept         2  164.2   82.11   1.679  0.204
exp          1   78.0   78.03   1.595  0.216
dept:exp     2   20.2   10.11   0.207  0.814
Residuals   30 1467.2   48.91               
> 
```

Practical-10

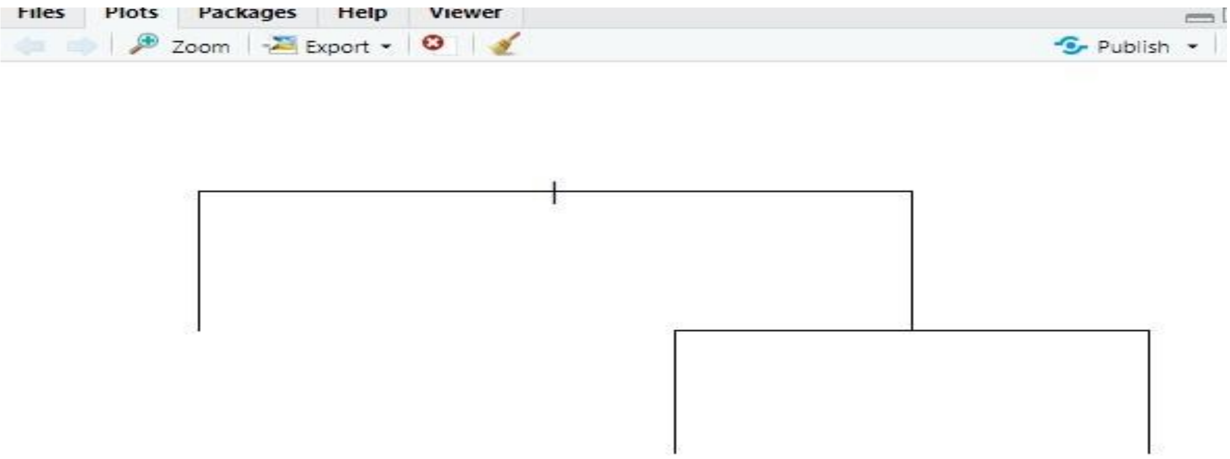
Aim: Practical of Decision Tree

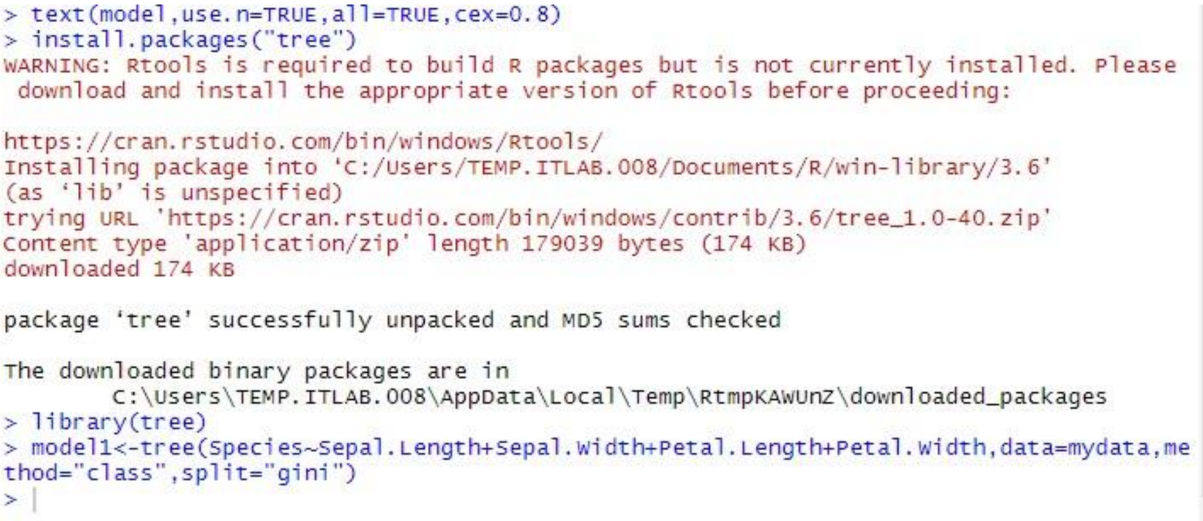
```
mydata<-data.frame(iris) attach(mydata)

install.packages("rpart") library(rpart) model<-

rpart(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,da ta=mydata,method="class")

plot(model) text(model,use.n=TRUE,all=TRUE,cex=0.8) install.packages("tree") library(tree) model1<-

tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data

=mydata,method="class",split="gini") plot(model1)

text(model,all=TRUE,cex=0.6) install.packages("party")

library(party) model2<-

ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,dat a=mydata) plot(model2)

library(tree) mydata<-data.frame(iris) attach(mydata)

model<tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data

=mydata,method="class",control=tree.control(nobs=150,mincut=10)) plot(model1)

text(model,all=TRUE,cex=0.6) predict(model,iris) model2<-

ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,dat

a=mydata,controls=ctree_control(maxdepth=2)) plot(model2)
```

Output :

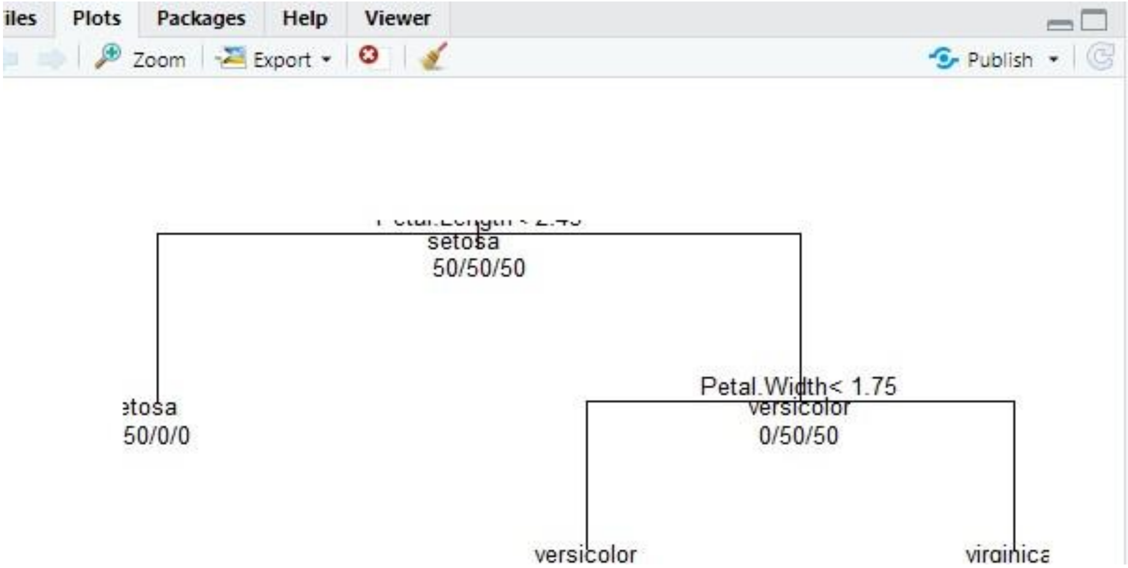


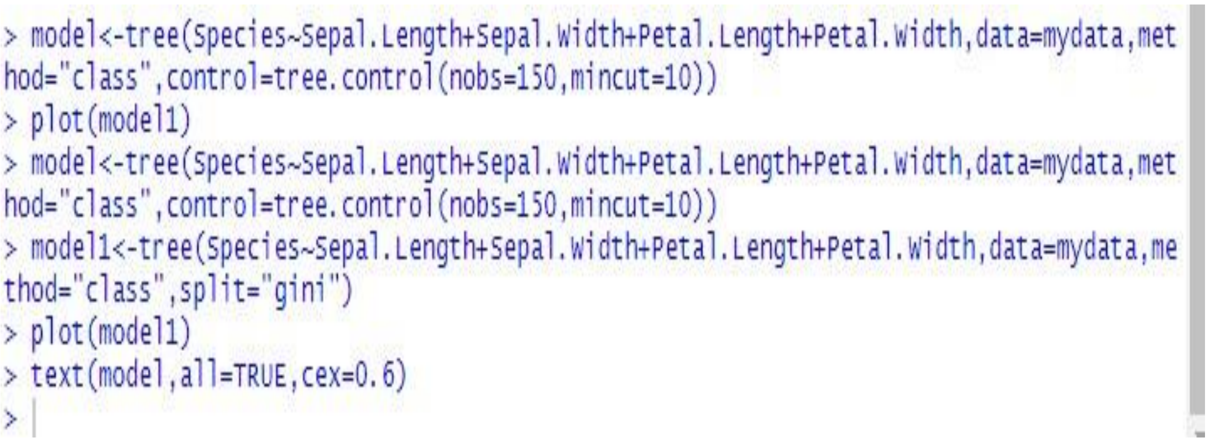
```
> mydata<-data.frame(iris)
> attach(mydata)
> install.packages("rpart")
WARNING: Rtools is required to build R packages but is not currently installed. Please do
wnload and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/TEMP.ITLAB.008/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/rpart_4.1-15.zip'
Content type 'application/zip' length 769889 bytes (751 KB)
downloaded 751 KB

package 'rpart' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\TEMP.ITLAB.008\AppData\Local\Temp\RtmpKAWUnZ\downloaded_packages
> library(rpart)
> model<-rpart(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,meth
od="class")
> plot(model)
> |
```



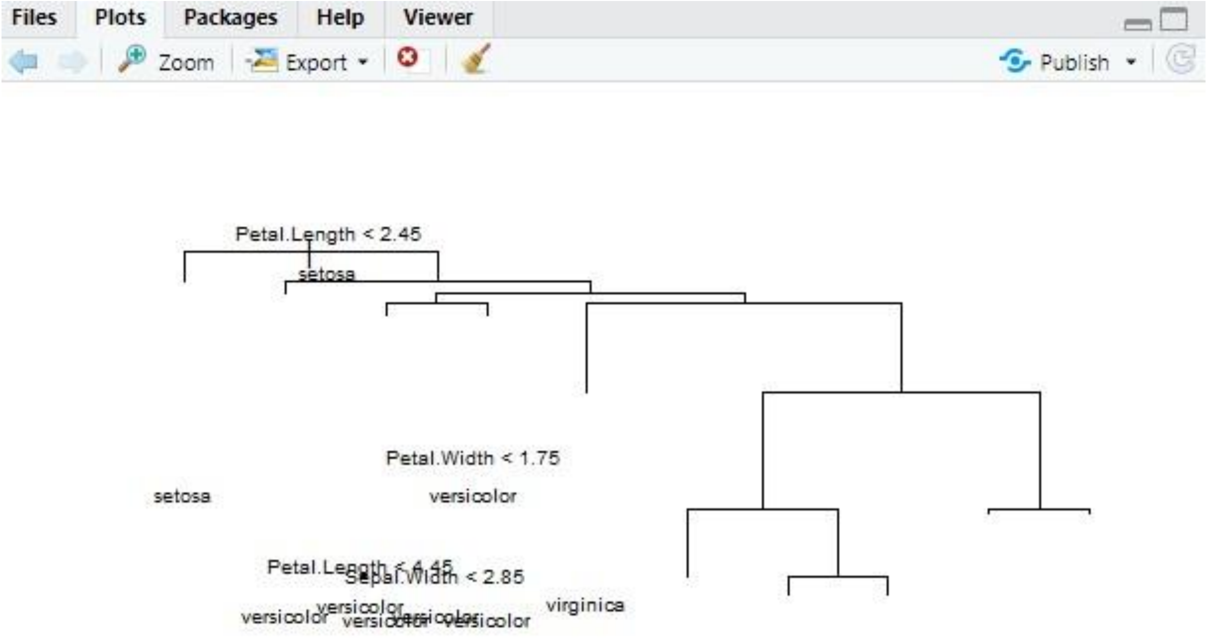


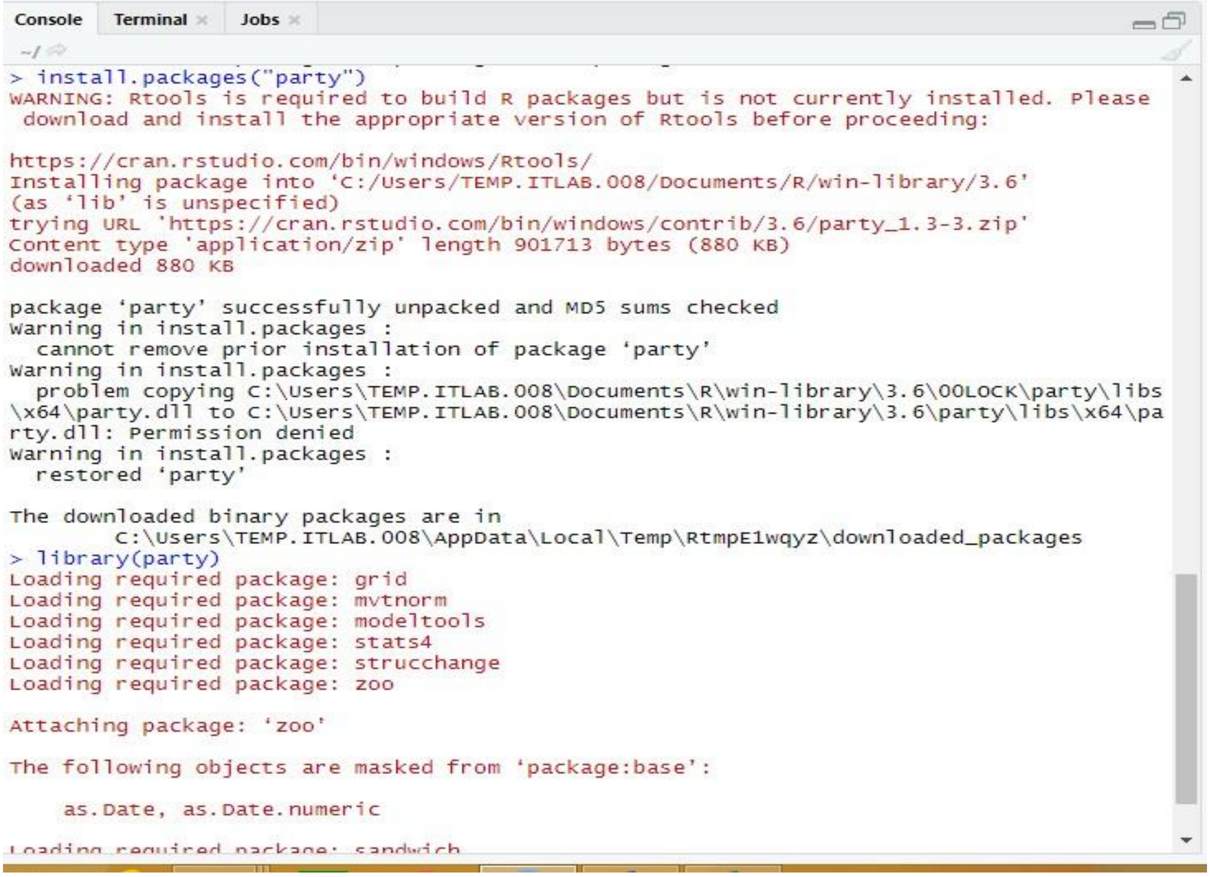
```
> text(model,use.n=TRUE,all=TRUE,cex=0.8)
> install.packages("tree")
WARNING: Rtools is required to build R packages but is not currently installed. Please
 download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/TEMP.ITLAB.008/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/tree_1.0-40.zip'
Content type 'application/zip' length 179039 bytes (174 KB)
downloaded 174 KB

package 'tree' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\TEMP.ITLAB.008\AppData\Local\Temp\RtmpKAWUnZ\downloaded_packages
> library(tree)
> model1<-tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,me
thod="class",split="gini")
> |
```



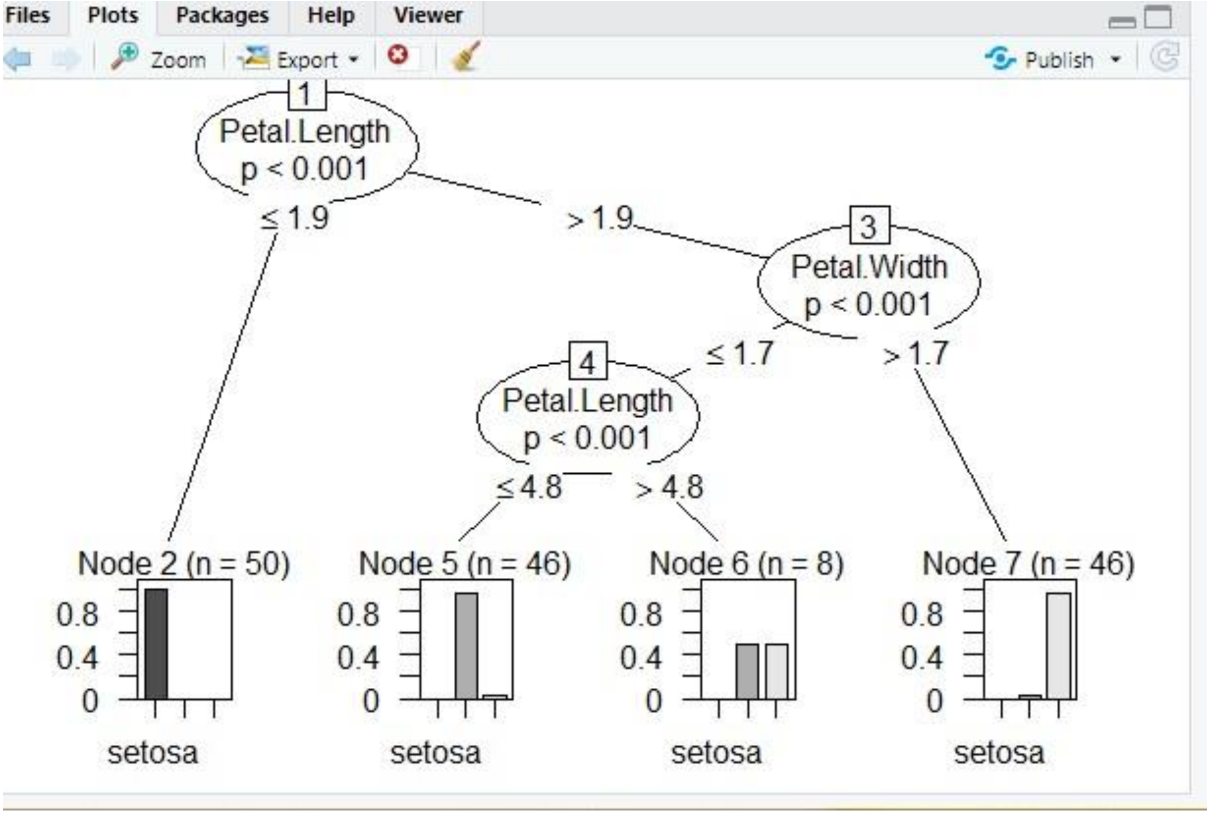


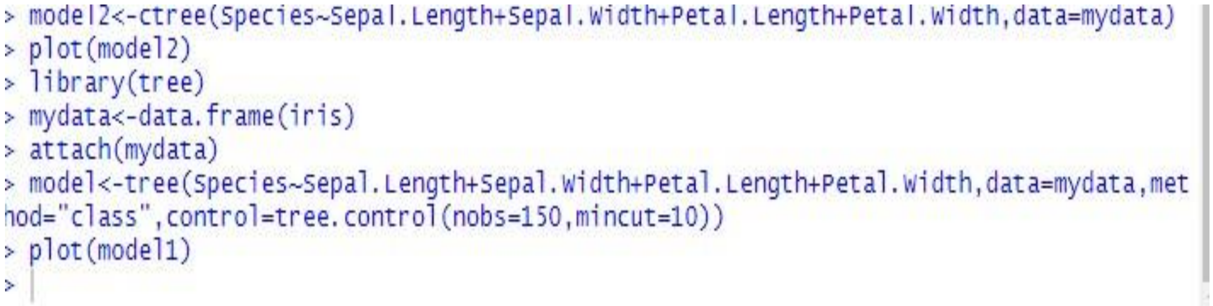
```
> model<-tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,met
hod="class",control=tree.control(nobs=150,mincut=10))
> plot(model1)
> model<-tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,met
hod="class",control=tree.control(nobs=150,mincut=10))
> model1<-tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,me
thod="class",split="gini")
> plot(model1)
> text(model,all=TRUE,cex=0.6)
>
```



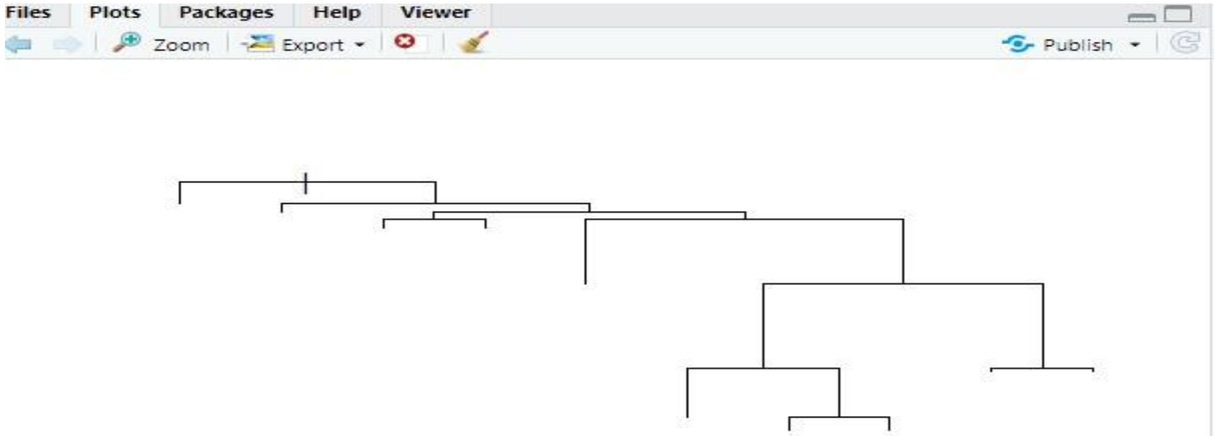


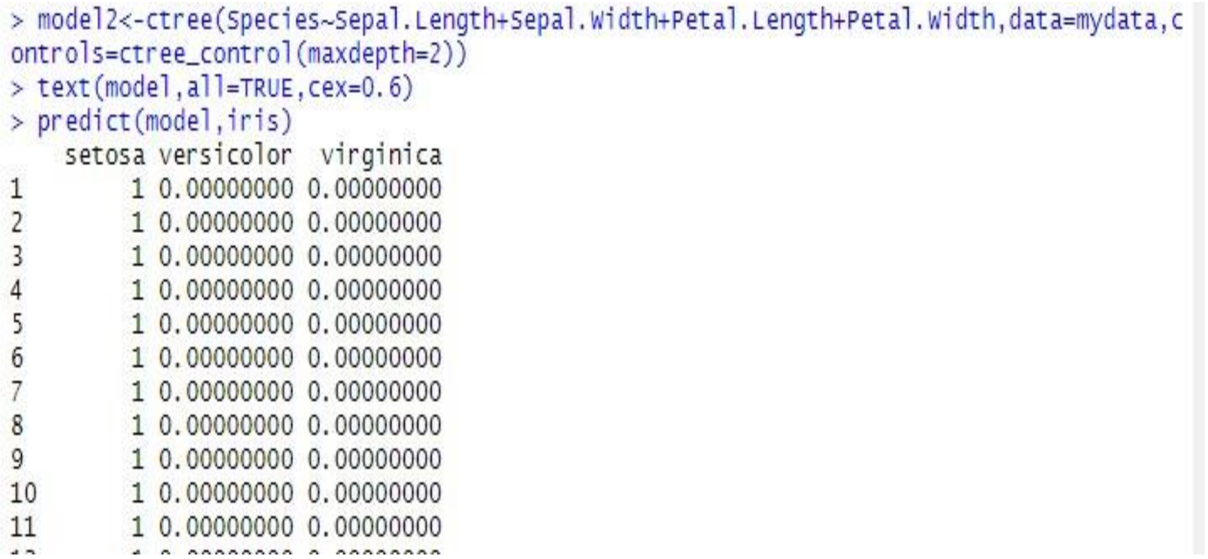
```
> install.packages("party")
WARNING: Rtools is required to build R packages but is not currently installed. Please
 download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/TEMP.ITLAB.008/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/party_1.3-3.zip'
Content type 'application/zip' length 901713 bytes (880 KB)
downloaded 880 KB

package ‘party’ successfully unpacked and MD5 sums checked
Warning in install.packages :
  cannot remove prior installation of package ‘party’
Warning in install.packages :
  problem copying C:\Users\TEMP.ITLAB.008\Documents\R\win-library\3.6\00LOCK\party\libs
\x64\party.dll to C:\Users\TEMP.ITLAB.008\Documents\R\win-library\3.6\party\libs\x64\pa
rty.dll: Permission denied
Warning in install.packages :
  restored ‘party’

The downloaded binary packages are in
	C:\Users\TEMP.ITLAB.008\AppData\Local\Temp\RtmpE1wqyz\downloaded_packages
> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: sandwich
```



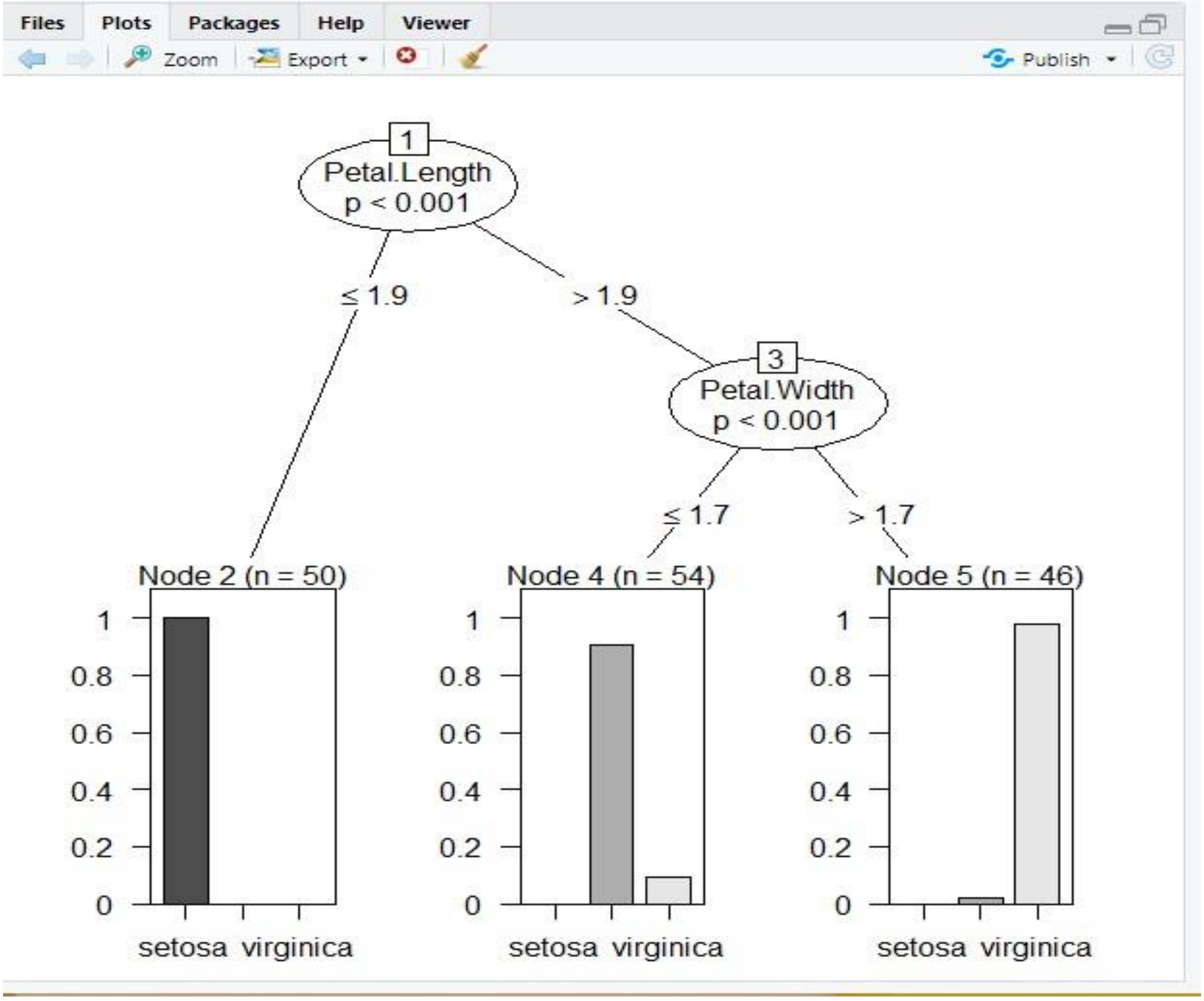


```
> model2<-ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata)
> plot(model2)
> library(tree)
> mydata<-data.frame(iris)
> attach(mydata)
> model<-tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,met
hod="class",control=tree.control(nobs=150,mincut=10))
> plot(model1)
> 
```





```
> model2<-ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,c
ontrols=ctree_control(maxdepth=2))
> text(model,all=TRUE,cex=0.6)
> predict(model,iris)
    setosa versicolor  virginica
1        1 0.00000000 0.00000000
2        1 0.00000000 0.00000000
3        1 0.00000000 0.00000000
4        1 0.00000000 0.00000000
5        1 0.00000000 0.00000000
6        1 0.00000000 0.00000000
7        1 0.00000000 0.00000000
8        1 0.00000000 0.00000000
9        1 0.00000000 0.00000000
10       1 0.00000000 0.00000000
11       1 0.00000000 0.00000000
```


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
1
Petal.Length
p < 0.001
≤ 1.9
> 1.9
3
Petal.Width
p < 0.001
≤ 1.7
> 1.7
Node 2 (n = 50)
Node 4 (n = 54)
Node 5 (n = 46)
1
0.8
0.6
0.4
0.2
0
setosa virginica