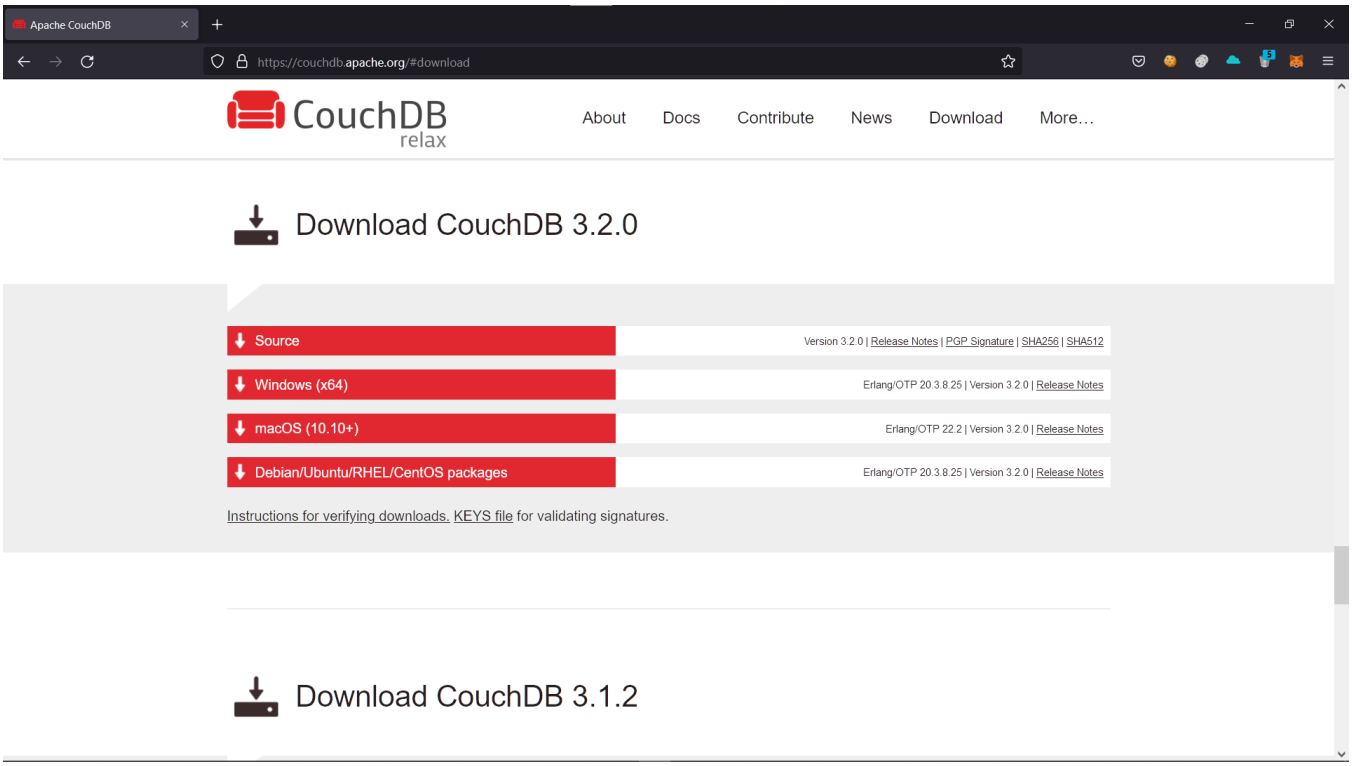
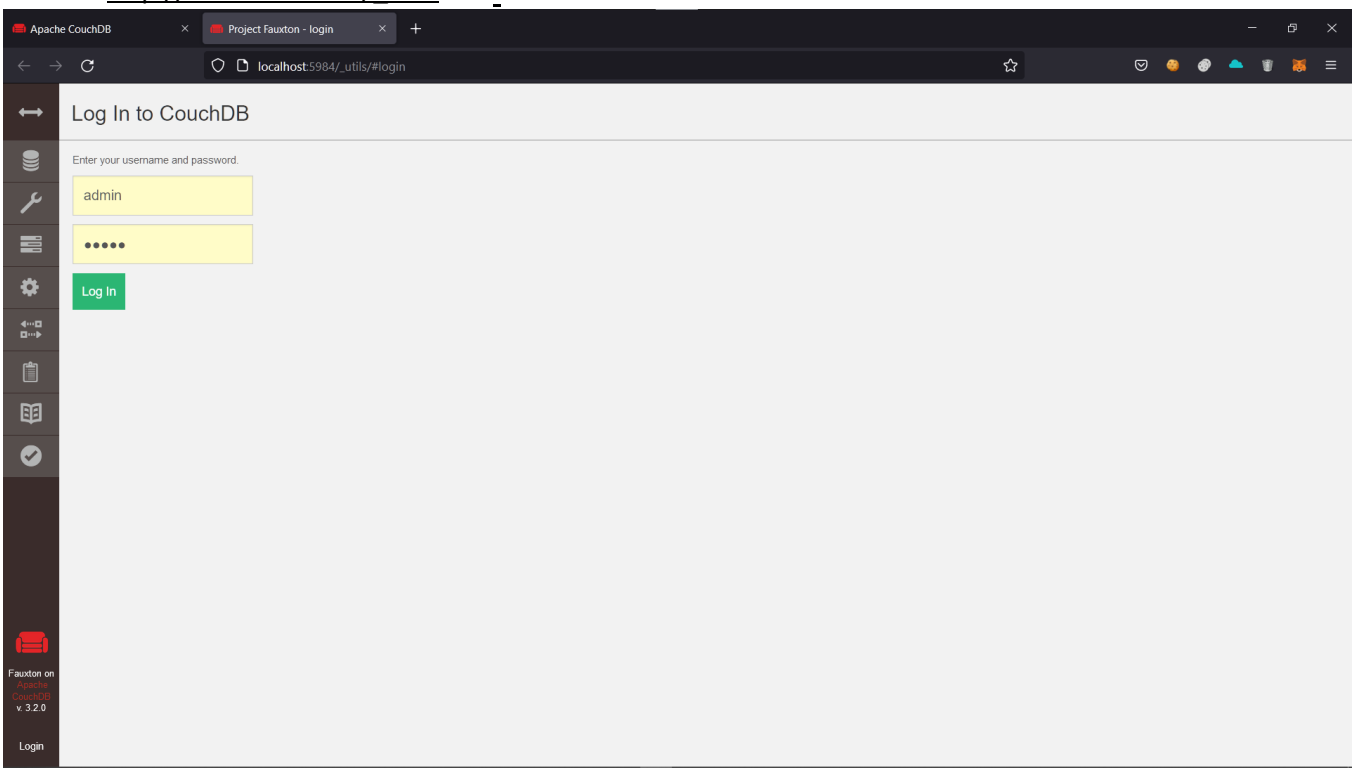
PRACTICAL NO. 1

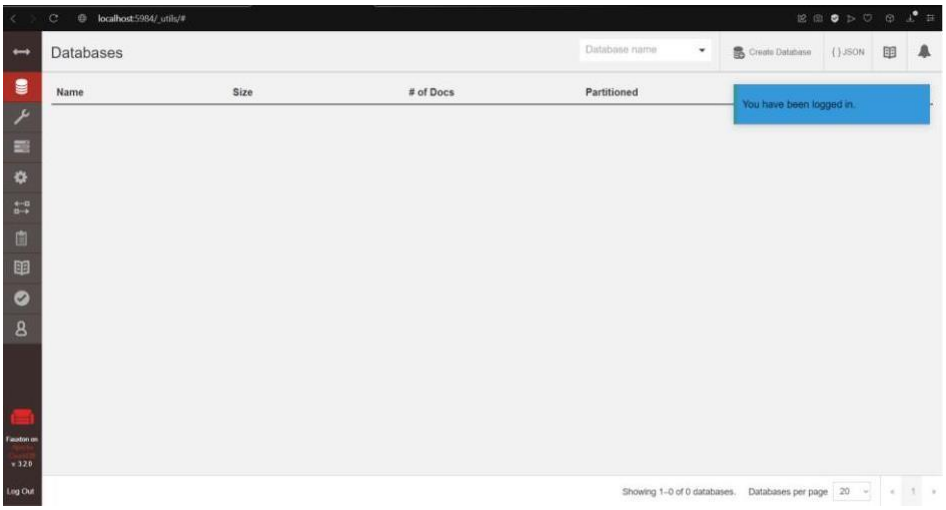
Aim: Practical of Data collection, Data curation and management for Unstructured data (NoSQL) Download and install CouchDB

Set up and validate new login credientials



Now visit http://localhost:5984/_utils and enter the credentials





Now open r console Install Package

```
> install.packages('sofa')
Installing package into 'C:/Users/GOD/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cloud.r-project.org/bin/windows/contrib/4.1/sofa_0.4.0.zip'
Content type 'application/zip' length 964993 bytes (942 KB)
downloaded 942 KB

package 'sofa' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\GOD\AppData\Local\Temp\RtmpKe2fGr\downloaded_packages
```

Connect to the database:

```
> library('sofa')
Warning message:
package 'sofa' was built under R version 4.1.3
> x<-Cushion$new()
> x$ping()
$couchdb
[1] "Welcome"

$version
[1] "3.2.0"

$git_sha
[1] "efb409bba"

$uuid
[1] "83591cbd6053a5614df052669260365d"

$features
$features[[1]]
[1] "access-ready"

$features[[2]]
[1] "partitioned"

$features[[3]]
[1] "pluggable-storage-engines"

$features[[4]]
[1] "reshard"

$features[[5]]
[1] "scheduler"


$vendor
$vendor$name
[1] "The Apache Software Foundation"

> x<-Cushion$new(user="admin",pwd='admin')
> db_create(x,dbname='ty')
$ok
[1] TRUE
```

Insert rows and commit changes:

```
> db_list(x)
[1] "ds"       "dssssss" "ty"
> doc1<-'{"rollno":"01","name":"ABC","GRADE":"A"}'
> doc_create(x,doc1,dbname="ds",docid="a_1")
$error
[1] "conflict"

$reason
[1] "Document update conflict."

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

$id
[1] "a_2"

$rev
[1] "3-2c77dd91f91632ebc5901a6b71dd25bd"

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

$reason
[1] "Document update conflict."

> db_changes(x,"ds")
$results
$results[[1]]
$results[[1]]$seq
[1] "4-g1AAAAB5eJzLYWBgYMpgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUklMiTV____PyuDOZE1Fy

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

$results[[1]]$changes
$results[[1]]$changes[[1]]
$results[[1]]$changes[[1]]$rev
[1] "2-1d6e4b76ed96977d5ecc10f4d7109166"
```

Query Database

```
[[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-lddcb45704c37893389b050ddbdc440a"

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

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

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


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

[[3]]$`_rev`
[1] "1-f13d2a583fc7fd0d645d421014a295b2"

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

```
[[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-lddcb45704c37893389b050ddbdc440a"

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

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

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

```
> db_query(x,dbname="ty",selector=list(REMARK="PASS"))$docs
[[1]]
[[1]]$`_id`
[1] "a_3"

[[1]]$`_rev`
[1] "1-f13d2a583fc7fd0d645d421014a295b2"

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

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

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

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


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

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



$bookmark
[1] "g1AAAAA2eJzLYWBgYMpgSmHgKy5JLCrJTg2MTI1lPzkzJBYozJ8Ybg6Q4YFIwwSwAiEQOcQ"

$warning
[1] "No matching index found, create an index to optimize query time."
```

Delete Record:

```
> library("jsonlite")
> res<-db_query(x,dbname="ty",selector=list('_id'=list('$gt'=NULL)),fields=c("name","rollno","GRADE","REMARK"),as="json")
>
> fromJSON(res)$docs
  name rollno GRADE REMARK
1  ABC     01     A   <NA>
2  PQR     02     A   <NA>
3  xyz     03     B   PASS
> doc_delete(x,dbname="ty",docid="a_2")
$ok
[1] TRUE

$id
[1] "a_2"

$rev
[1] "2-82f1879cc7d73bef5574cc5cdf7c4094"
```

Update a record:

```
> doc_get(x,dbname = "ty",docid = "a_2")
Error: (404) - deleted
> doc2<-'{"name":"$food","biryani":"TEST","note":"yummy","note2":"yay"}'
> doc_update(x,dbname = "ty",doc=doc2,docid="a_3",rev = "3-b1fb56db955b142c6efd3b3c52fe9e1b")
Error: (409) - Document update conflict.
> doc_update(x,dbname="ty",doc=doc3,docid="a_3",rev="1-683f5507333722ba7596bf4ad21635ad")
$ok
[1] TRUE

$id
[1] "a_3"

$rev
[1] "2-a44a16914e2078d5ff784b4f7633c181"

> doc3<-'{"rollno":"01",
+ "name":"UZMA",
+ "GRADE":"A"}'
> doc_update(x,dbname = "ds",doc=doc3,docid = "a_1",rev = "1-be7c98bddf8ea7c46f4f401ff387593d")
Error: (404) - Database does not exist.
> doc_update(x,dbname = "ty",doc=doc3,docid = "a_1",rev = "1-be7c98bddf8ea7c46f4f401ff387593d")
$ok
[1] TRUE

$id
[1] "a_1"

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

PRACTICAL NO. 2

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

```
C:\Program Files\MongoDB\Server\5.0\bin>mongo
MongoDB shell version v5.0.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("daf9c94b-f37f-4bf1-8734-3eee52916846") }
MongoDB server version: 5.0.5
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-03-19T12:37:34.239+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

Show databases

```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
```

Switch to beginners_book (collections)

```
> db
test
> use beginners_book
switched to db beginners_boo
```

Insert records

```
> db.beginners_book.insert({name:"nick",age:20,website:"codewithnick.github.io"})
WriteResult({ "nInserted" : 1 })
> db.beginners_book.insert({name:"nikhil",age:19,website:"www.fb.com"})
WriteResult({ "nInserted" : 1 })
> db.beginners_book.insert({name:"nikhilsingh",age:20,website:"www.ig.com"})
WriteResult({ "nInserted" : 1 })
> db.beginners_book.insert({name:"nikhil",age:21,website:"www.codewithnick.linktree.com"})
WriteResult({ "nInserted" : 1 })
>
```

Look at records

```
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nick", "age" : 20, "website" : "codewithnick.github.io" }
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website" : "www.fb.com" }
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "website" : "www.ig.com" }
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website" : "www.codewithnick.linktree.com" }
>
```

Insert dynamic records

```
> db.beginners_book.insert({ name:"john", age:35, website:"www.fb.com",email:"admin@beginnersbooks.com",course:[{name:"MongoDB",duration:7},{name:"Java",duration:14}] })
WriteResult({ "nInserted" : 1 })
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nick", "age" : 20, "website" : "codewithnick.github.io" }
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website" : "www.fb.com" }
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "website" : "www.ig.com" }
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website" : "www.codewithnick.linktree.com" }
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "john", "age" : 35, "website" : "www.fb.com", "email" : "admin@beginnersbooks.com", "course" : [ { "name" : "MongoD
B", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
>
```

Create, view and drop collections

```
> show collections
beginners_book
> db.createCollection("students")
{ "ok" : 1 }
> db.createCollection("teachers")
{ "ok" : 1 }
> show collections
beginners_book
students
teachers
> db.teachers.drop()
true
> show collections
beginners_book
students
> _
```

Bulk insert

```
> var beginners=[{"StudentId":1001,"StudentName":"Steve","age":30},{"StudentId":1002,"StudentName":"Negan","age":30},{"StudentId":1003,"StudentName":"Rick","age":31}];
> db.students.insert(beginners);
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 3,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.students.find()
{ "_id" : ObjectId("6236c1c96013afad8f7dafd1"), "StudentId" : 1001, "StudentName" : "Steve", "age" : 30 }
{ "_id" : ObjectId("6236c1c96013afad8f7dafd2"), "StudentId" : 1002, "StudentName" : "Negan", "age" : 30 }
{ "_id" : ObjectId("6236c1c96013afad8f7dafd3"), "StudentId" : 1003, "StudentName" : "Rick", "age" : 31 }
```

View each record

```
> db.students.find().forEach(printjson)
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd1"),
        "StudentId" : 1001,
        "StudentName" : "Steve",
        "age" : 30
}
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd2"),
        "StudentId" : 1002,
        "StudentName" : "Negan",
        "age" : 30
}
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd3"),
        "StudentId" : 1003,
        "StudentName" : "Rick",
        "age" : 31
}
>
```

Search records according to attributes

```
> db.students.find({StudentName:"Steve"}).pretty()
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd1"),
        "StudentId" : 1001,
        "StudentName" : "Steve",
        "age" : 30
}
> db.students.find({"age":{$gt:30}}).pretty()
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd3"),
        "StudentId" : 1003,
        "StudentName" : "Rick",
        "age" : 31
}
> db.students.find({"StudentId":{$lt:1003}}).pretty()
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd1"),
        "StudentId" : 1001,
        "StudentName" : "Steve",
        "age" : 30
}
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd2"),
        "StudentId" : 1002,
        "StudentName" : "Negan",
        "age" : 30
}
> db.students.find({"StudentName":{$ne:"Negan"}}).pretty()
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd1"),
        "StudentId" : 1001,
        "StudentName" : "Steve",
        "age" : 30
}
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd3"),
        "StudentId" : 1003,
        "StudentName" : "Rick",
        "age" : 31
}
>
```

Update query

```
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nick", "age" : 20, "website"
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "we
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "john", "age" : 35, "website"
B", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
> db.beginners_book.update({"name":"john"},{$set:{"name":"nick"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nick", "age" : 20, "website"
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "we
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "nick", "age" : 35, "website"
B", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
```

```
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nick", "age" : 20, "website" : "codewithnick.github.io" }
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website" : "www.fb.com" }
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "website" : "www.ig.com" }
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website" : "www.codewithnick.linktree.com" }
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "nick", "age" : 35, "website" : "www.fb.com", "email" : "admin@begin
B", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
> db.beginners_book.update({"name":"nick"},{$set:{"name":"nikhilnew"}},{multi:true})
WriteResult({ "nMatched" : 2, "nUpserted" : 0, "nModified" : 2 })
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nikhilnew", "age" : 20, "website" : "codewithnick.github.io" }
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website" : "www.fb.com" }
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "website" : "www.ig.com" }
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website" : "www.codewithnick.linktree.com" }
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "nikhilnew", "age" : 35, "website" : "www.fb.com", "email" : "admin@
ongoDB", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
```

Save if not exists or update if exists

```
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nikhilnew", "age" : 20, "website" : "codewithnick.github.io
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website" : "www.fb.com" }
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "website" : "www.ig.com" }
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website" : "www.codewithnick.linktree
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "nikhilnew", "age" : 35, "website" : "www.fb.com", "email" :
ongoDB", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
> db.beginners_book.save({"_id" : ObjectId("61e6232115344ea035aebeff"),"name":"John","age":25})
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 1,
        "nModified" : 0,
        "_id" : ObjectId("61e6232115344ea035aebeff")
})
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nikhilnew", "age" : 20, "website" : "codewithnick.github.io
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19, "website" : "www.fb.com" }
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age" : 20, "website" : "www.ig.com" }
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21, "website" : "www.codewithnick.linktree
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "nikhilnew", "age" : 35, "website" : "www.fb.com", "email" :
ongoDB", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
{ "_id" : ObjectId("61e6232115344ea035aebeff"), "name" : "John", "age" : 25 }
>
```

Remove

```
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nikhilnew", "age" :
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age"
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "nikhilnew", "age" :
ongoDB", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
{ "_id" : ObjectId("61e6232115344ea035aebeff"), "name" : "John", "age" : 25 }
> db.beginners_book.remove({"name":"John"})
WriteResult({ "nRemoved" : 1 })
> db.beginners_book.find()
{ "_id" : ObjectId("6236be416013afad8f7dafcb"), "name" : "nikhilnew", "age" :
{ "_id" : ObjectId("6236be5e6013afad8f7dafcc"), "name" : "nikhil", "age" : 19
{ "_id" : ObjectId("6236be6c6013afad8f7dafcd"), "name" : "nikhilsingh", "age"
{ "_id" : ObjectId("6236be8c6013afad8f7dafce"), "name" : "nikhil", "age" : 21
{ "_id" : ObjectId("6236c0d16013afad8f7dafd0"), "name" : "nikhilnew", "age" :
ongoDB", "duration" : 7 }, { "name" : "Java", "duration" : 14 } ] }
>
```

Finding using more attrs

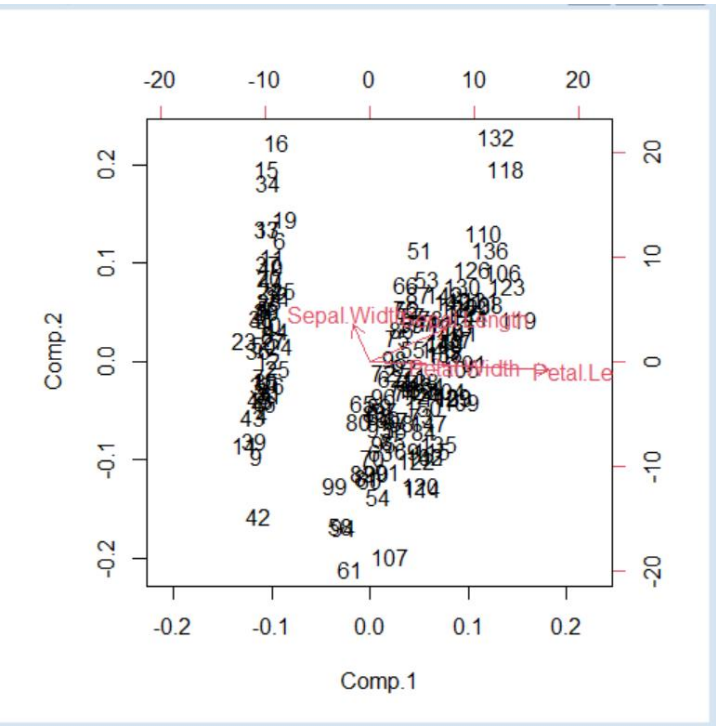
```
> db.students.find()
{ "_id" : ObjectId("6236c1c96013afad8f7dafd1"), "StudentId" : 1001, "StudentName" : "Steve", "age" : 30 }
{ "_id" : ObjectId("6236c1c96013afad8f7dafd2"), "StudentId" : 1002, "StudentName" : "Negan", "age" : 30 }
{ "_id" : ObjectId("6236c1c96013afad8f7dafd3"), "StudentId" : 1003, "StudentName" : "Rick", "age" : 31 }
> db.students.find({},{"_id":0,"StudentId":1})
{ "StudentId" : 1001 }
{ "StudentId" : 1002 }
{ "StudentId" : 1003 }
> db.students.find({},{"_id":0,"StudentName":0,"age":0})
{ "StudentId" : 1001 }
{ "StudentId" : 1002 }
{ "StudentId" : 1003 }
> db.students.find({"StudentId":{$gt:1001}}).pretty()
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd2"),
        "StudentId" : 1002,
        "StudentName" : "Negan",
        "age" : 30
}
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd3"),
        "StudentId" : 1003,
        "StudentName" : "Rick",
        "age" : 31
}
> db.students.find({"StudentId":{$gt:1001}}).limit(1).pretty()
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd2"),
        "StudentId" : 1002,
        "StudentName" : "Negan",
        "age" : 30
}
```

```
> db.students.find({"StudentId":{$gt:1001}}).limit(1).skip(1).pretty()
{
        "_id" : ObjectId("6236c1c96013afad8f7dafd3"),
        "StudentId" : 1003,
        "StudentName" : "Rick",
        "age" : 31
}
> db.students.find({},{"_id":0,"StudentId":1}).sort({"StudentId":-1})
{ "StudentId" : 1003 }
{ "StudentId" : 1002 }
{ "StudentId" : 1001 }
> db.students.find({},{"_id":0,"StudentId":1}).sort({"StudentId":1})
{ "StudentId" : 1001 }
{ "StudentId" : 1002 }
{ "StudentId" : 1003 }
>
```

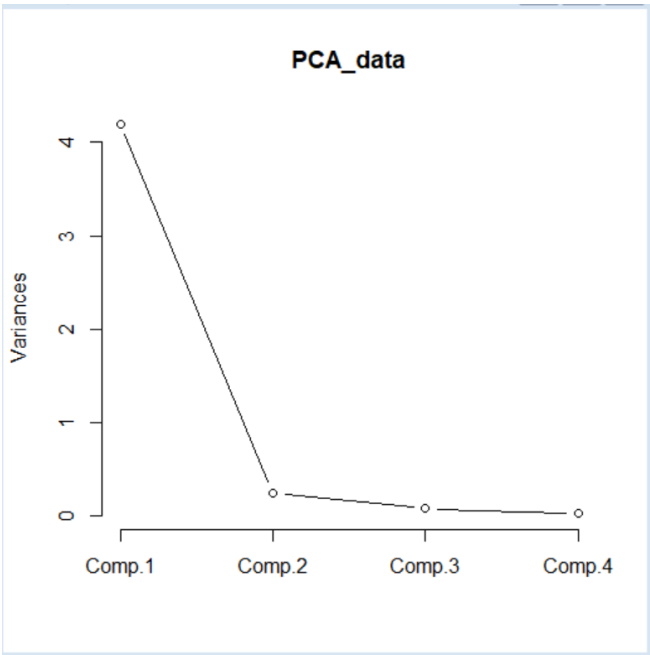
PRACTICAL NO. 3

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$values
[1] 4.22824171 0.24267075 0.07820950 0.02383509
> PCA_data$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)
>
```



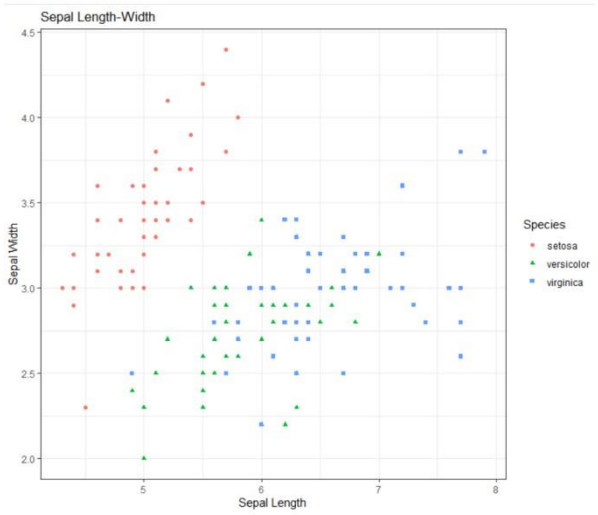
```
> screeplot(PCA_data, type="lines")
>  model2 = PCA_data$loadings[,1]
>  model2_scores <- as.matrix(data_iris) %*% model2
> library(class)
> install.packages("e1071")
Installing package into 'C:/Users/nikhi/OneDrive/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cloud.r-project.org/bin/windows/contrib/4.1/e1071_1.7-9.zip'
Content type 'application/zip' length 1022973 bytes (998 KB)
downloaded 998 KB

package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\nikhi\AppData\Local\Temp\RtmpSeqpae\downloaded_packages
> library(e1071)
Warning message:
package 'e1071' was built under R version 4.1.3
> 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
>
>
```

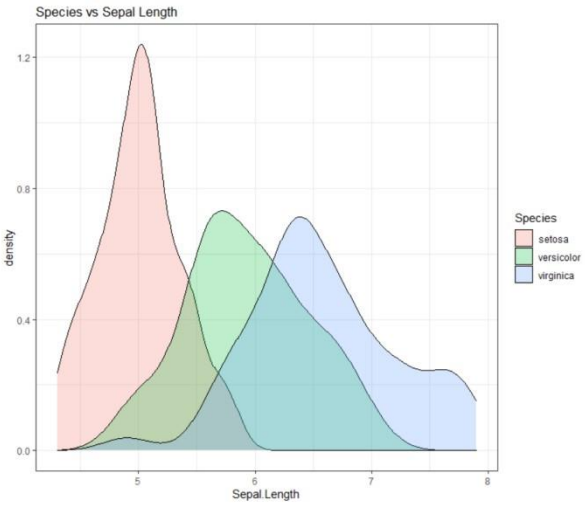


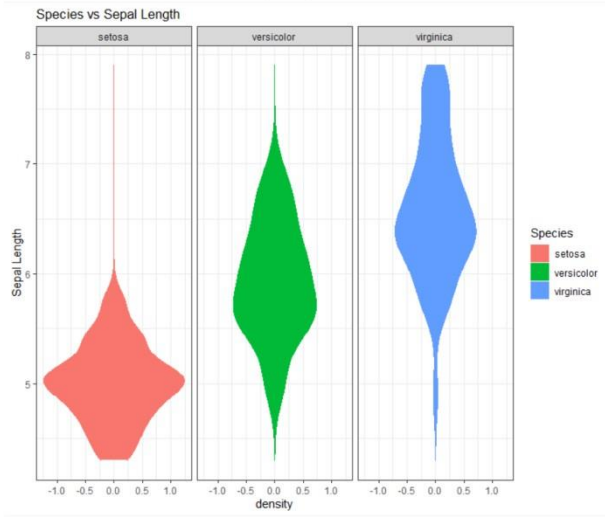
PRACTICAL NO. 4

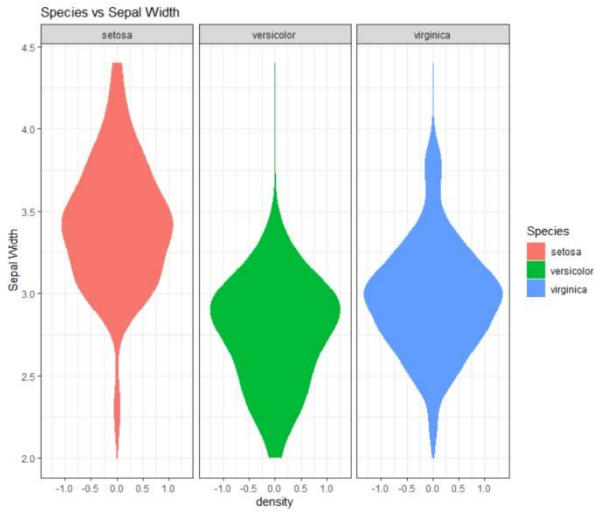
Aim: Practical of Clustering

```
> library(ggplot2)
> scatter <- ggplot(data=iris, aes(x = Sepal.Length, y = Sepal.Width))
> scatter
> scatter + geom_point(aes(color=Species, shape=Species)) +   theme_bw()+   xlab("Sepal Length") +  ylab("Sepal Width") +   ggtitle("Sepal Length-Width")
> ggplot(data=iris, aes(Sepal.Length, fill = Species))+    theme_bw()+   geom_density(alpha=0.25)+   labs(x = "Sepal.Length", title="Species vs Sepal Length"))
Error: unexpected ')' in "ggplot(data=iris, aes(Sepal.Length, fill = Species))+    theme_bw()+   geom_density(alpha=0.25)+   labs(x = "Sepal.Length", title="Species vs Sepal Length"))"
> ggplot(data=iris, aes(Sepal.Length, fill = Species))+    theme_bw()+   geom_density(alpha=0.25)+   labs(x = "Sepal.Length", title="Species vs Sepal Length")
> vol <- ggplot(data=iris, aes(x = Sepal.Width))
> vol + stat_density(aes(ymax = ..density..,  ymin = -..density..,fill = Species,color = Species),                    geom = "ribbon", position = "identity") +   facet_grid(. ~ Species) $
> vol <- ggplot(data=iris, aes(x = Sepal.Length))
> vol + stat_density(aes(ymax = ..density..,  ymin = -..density..,fill = Species, color = Species),geom = "ribbon", position = "identity") +   facet_grid(. ~ Species) + coord_flip() + the$
> #clustering method1
> irisData <- iris[,1:4]
> totalwSS<-c()
```

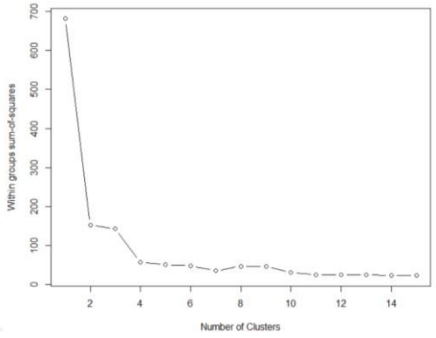




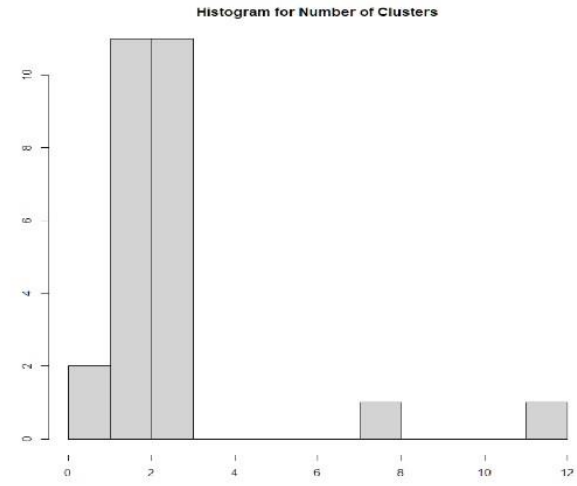


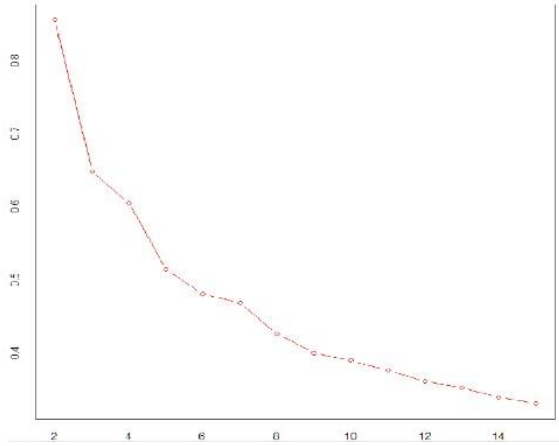


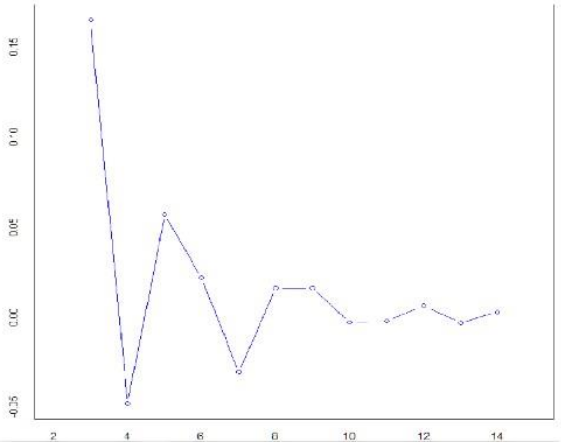

```
> #clustering method1
> irisData <- iris[,1:4]
> totalwSS<-c()
> for (i in 1:15)
+  {
+ clusterIRIS <- kmeans(irisData, centers=i)
+ totalwSS[i]<-clusterIRIS$tot.withinss
+ }
> plot(x=1:15,y=totalwSS,type="b",xlab="Number of Clusters",
+ ylab="Within groups sum-of-squares")
>
```



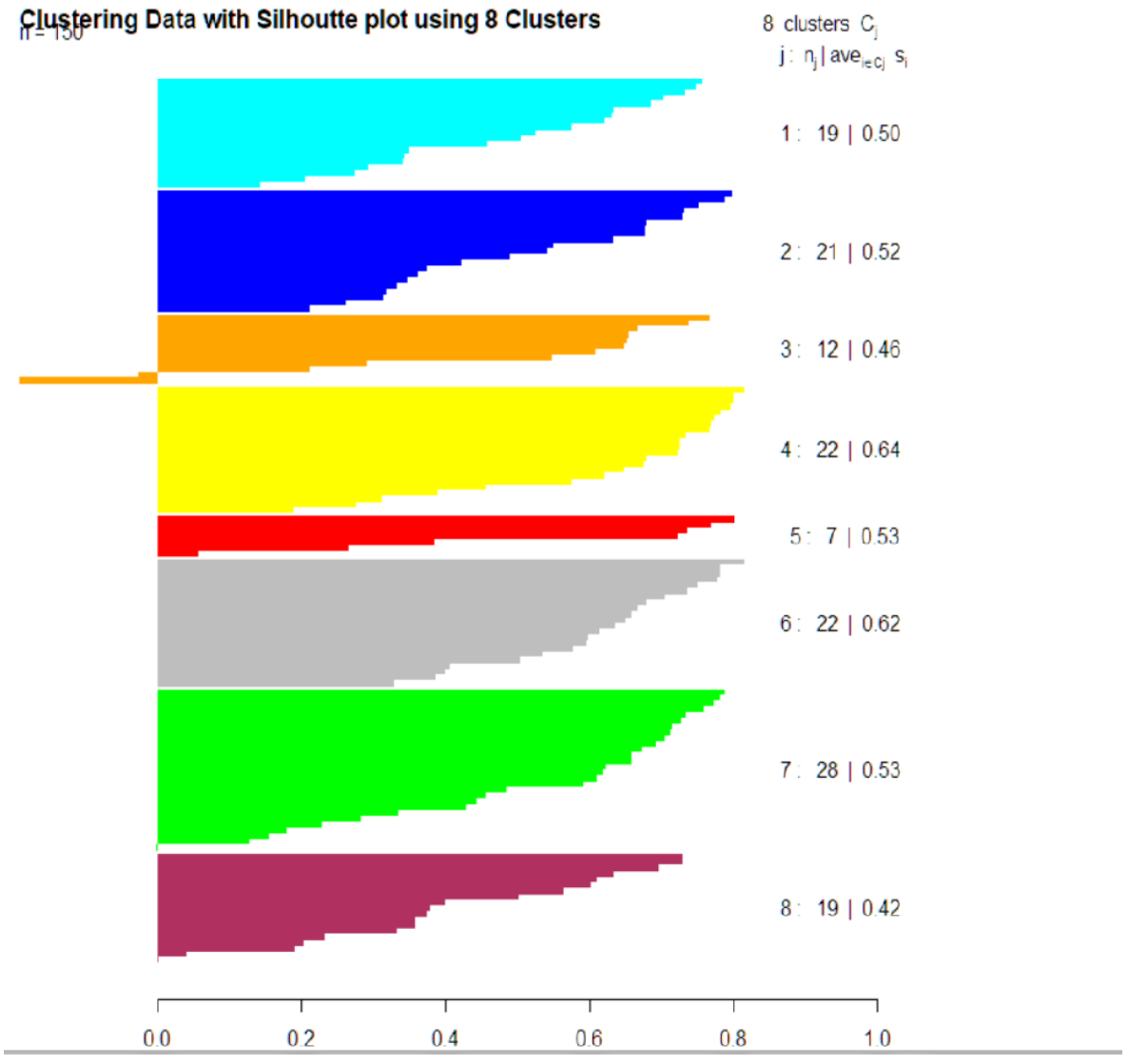
```
> install.packages("NbClust")
Installing package into 'C:/Users/GOD/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
trying URL 'https://cran.csiro.au/bin/windows/contrib/4.1/NbClust_3.0.zip'
Content type 'application/zip' length 122440 bytes (119 KB)
downloaded 119 KB

package 'NbClust' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\GOD\AppData\Local\Temp\RtmpO2ghgq\downloaded_packages
> library(NbClust)
> par(mar = c(2,2,2,2))
> nb <- NbClust(irisData, method = "kmeans")
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds to a
                significant increase of the value of the measure i.e the significant peak in Hubert
                index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
                In the plot of D index, we seek a significant knee (the significant peak in Dindex
                second differences plot) that corresponds to a significant increase of the value of
                the measure.

*******************************************************************
* Among all indices:
* 11 proposed 2 as the best number of clusters
* 11 proposed 3 as the best number of clusters
* 1 proposed 8 as the best number of clusters
* 1 proposed 12 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  2


*******************************************************************
> hist(nb$Best.nc[1,], breaks = 15, main="Histogram for Number of Clusters")
```







```
> library (clustertend)
Package `clustertend` is deprecated.  Use package `hopkins` instead.
> genx<-function (x) {
+ runif (length (x), min (x), (max (x)))
+ }
> random_df <-apply (irist, -51, 2, genx)
Error in match.fun(FUN) : '2' is not a function, character or symbol
> random_df <-apply (irist, -5], 2, genx)
Error: unexpected ']' in "random_df <-apply (irist, -5]"
> random_df <-apply (iris, -5], 2, genx)
Error: unexpected ']' in "random_df <-apply (iris, -5]"
> random_df <-apply (iris[, -5], 2, genx)
> install.packages ("factoextra")
Warning: package 'factoextra' is in use and will not be installed
> genx<-function (x) {
+ runif (length(x), min (x), (max (x)))}
> genx<-function (x) (runif (length(x), min (x), max (x) ) )
> genx<-function (x) (runif (length(x), min (x), max (x) ) )
> library(cluster)
> cl<-kmeans(iris[,-5],2)
> dis<-dist(iris[,5])^2
Warning message:
In dist(iris[, 5]) : NAs introduced by coercion
> dis<-dist(iris[,-5])^2
> sil=silhouette(cl$cluster,dis)
> plot(sil,main="clustering data with silhouette",col=c("cyan","blue"))
> cl<-kmeans(iris[,-5],8)
> dis<-dist(iris[,-5])^2
> sil=silhouette(cl$cluster,dis)
>  plot(sil, main = "Clustering Data with Silhoutte plot using 8 Clusters"$
```



PRACTICAL NO. 5

Aim: Practical of Time-series forecasting

#load Data Airpassenger

data("AirPassengers")

#finding the class name

class(AirPassengers)

#Data In time series format

#start of time series

start(AirPassengers)

#Exp:05

#start of time series

end(AirPassengers)

frequency(AirPassengers)

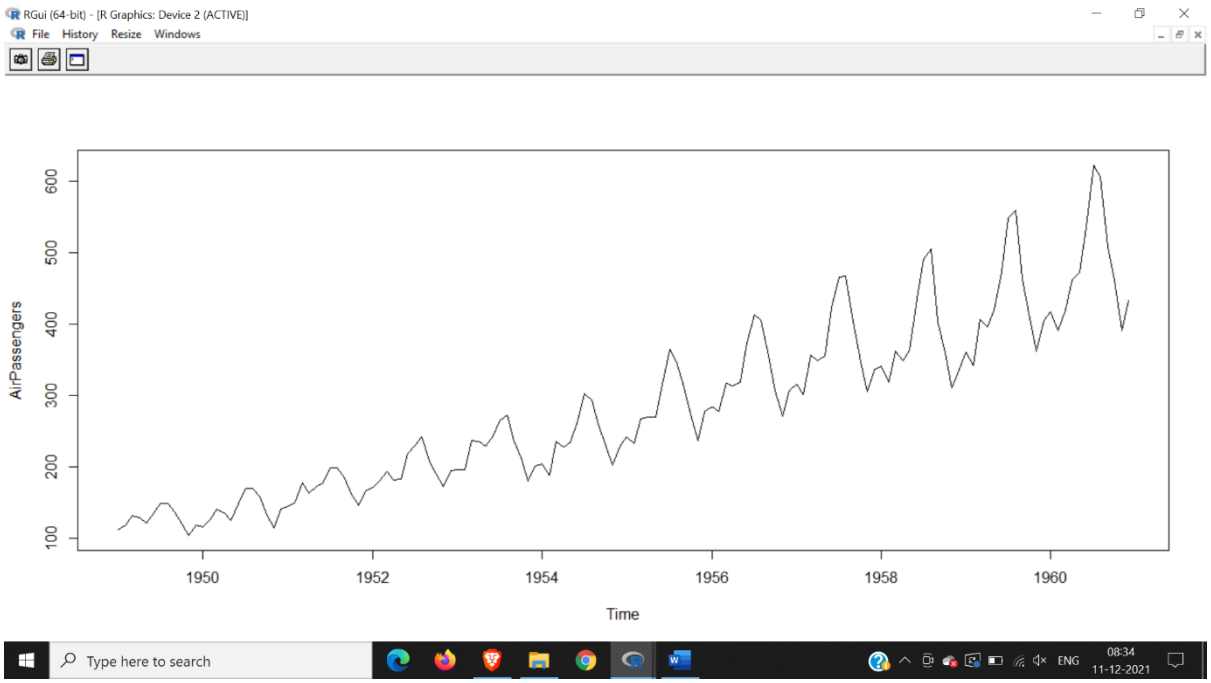
#The cycle of this time series is 12 month in a year

summary(AirPassengers)

```
[Previously saved workspace restored]

> #load Data Airpassenger
> data("AirPassengers")
> #finding the class name
> class(AirPassengers)
[1] "ts"
> #Data In time series format
> #start of time series
> start(AirPassengers)
[1] 1949    1
> #Exp:05
> #start of time series
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> #The cycle of this time series is 12 month in a year
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  104.0   180.0   265.5   280.3   360.5   622.0
>
```

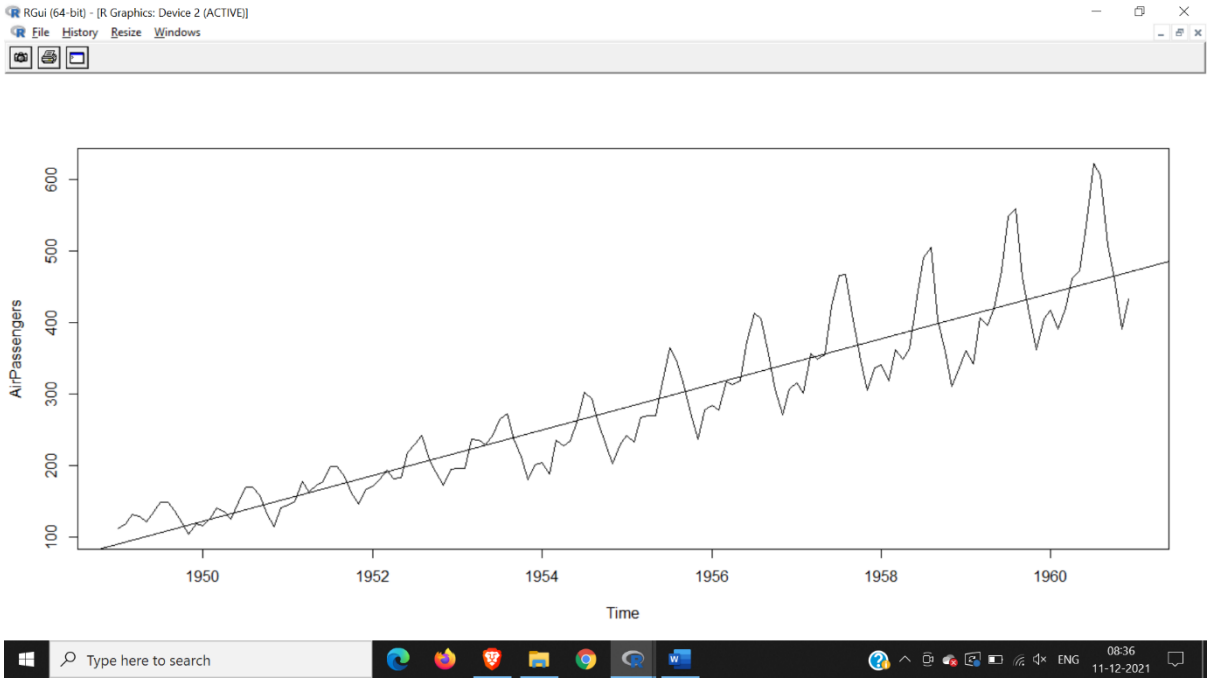
#the number of passenger are distributed across the spectrum

plot(AirPassengers)



#this will plot the time series

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



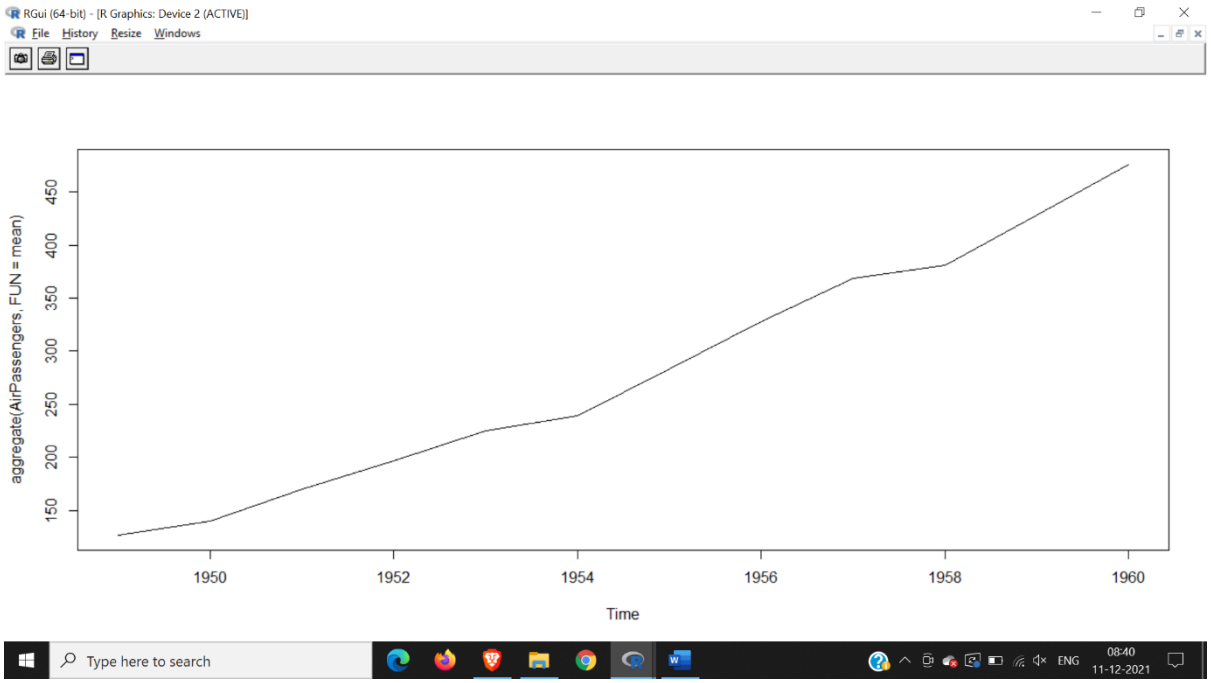
#This will print the cycle across years

cycle(AirPassengers)

```
RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help
[1] "ts"
> #Data In time series format
> #start of time series
> start(AirPassengers)
[1] 1949    1
> #Exp:05
> #start of time series
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> #The cycle of this time series is 12 month in a year
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  104.0   180.0   265.5   280.3   360.5   622.0
> #the number of passenger are distributed across the spectrum
> plot(AirPassengers)
> #this will plot the time series
> abline(reg=lm(AirPassengers~time(AirPassengers)))
> #This will print the cycle across years
> 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
>
Type here to search
ENG 08:37 11-12-2021
```

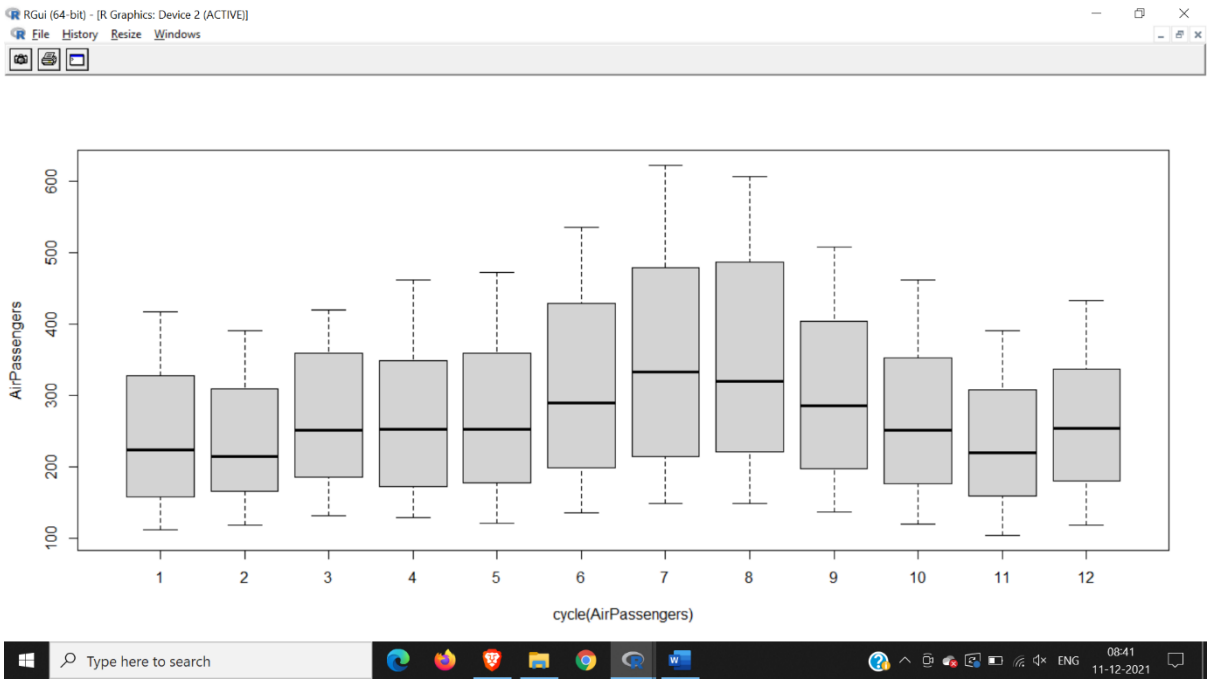
#this wil aggreate the cycles and display a year on year trend

plot(aggregate(AirPassengers,FUN=mean))

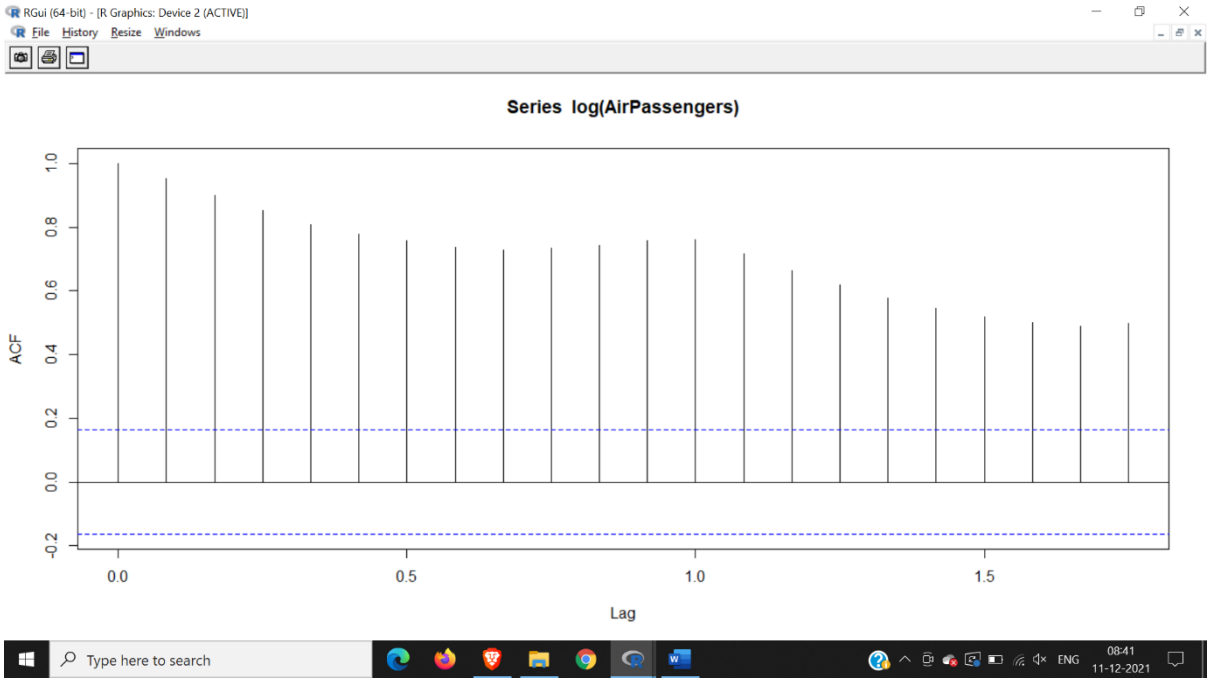


#box plot across month will give us a sense on seasonal effect

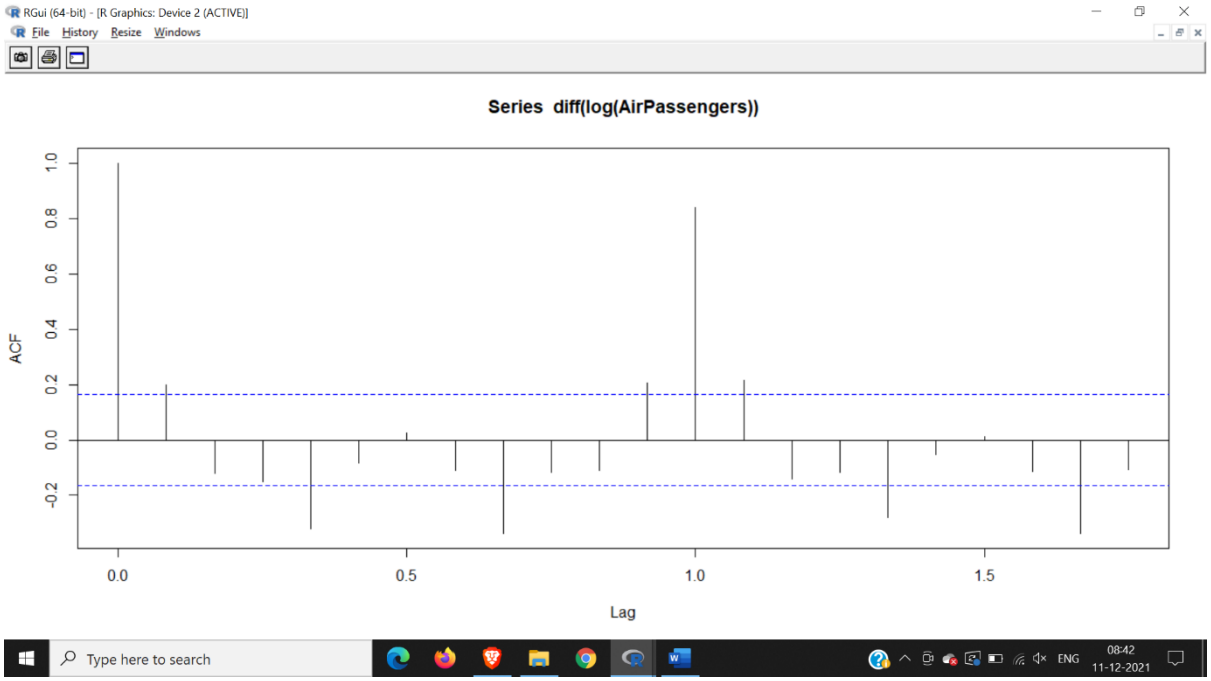
boxplot(AirPassengers~cycle(AirPassengers))



acf(log(AirPassengers))



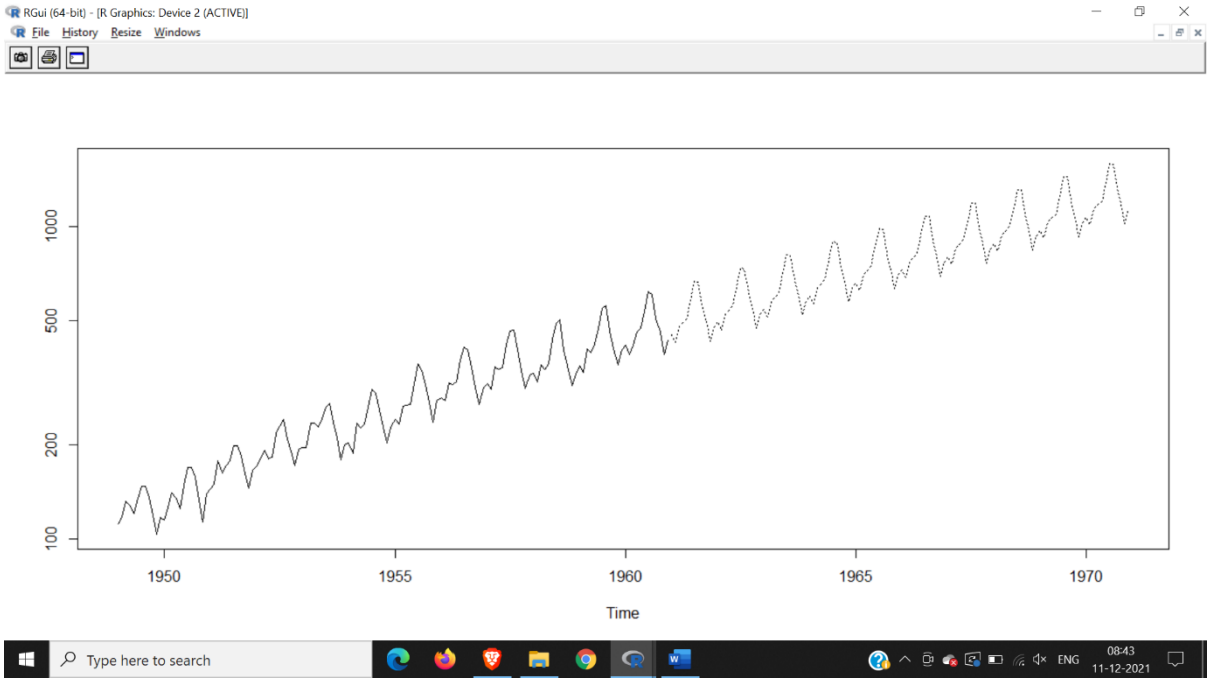
acf(diff(log(AirPassengers)))



(fit <- arima (log(AirPassengers), c(0, 1, 1), seasonal = list (order= c(0, 1,1),period=12)))

pred <- predict(fit, n.ahead = 10*12)

ts.plot(AirPassengera,2.718^pred$pred,log="y",lty=c(1,3))



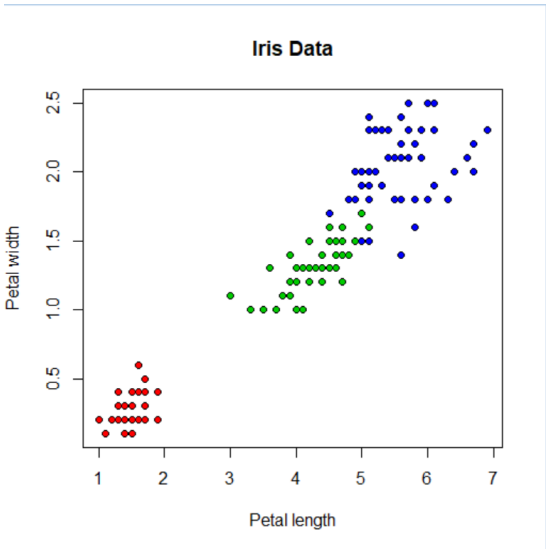
Hence we conclude there will be a rise in passengers

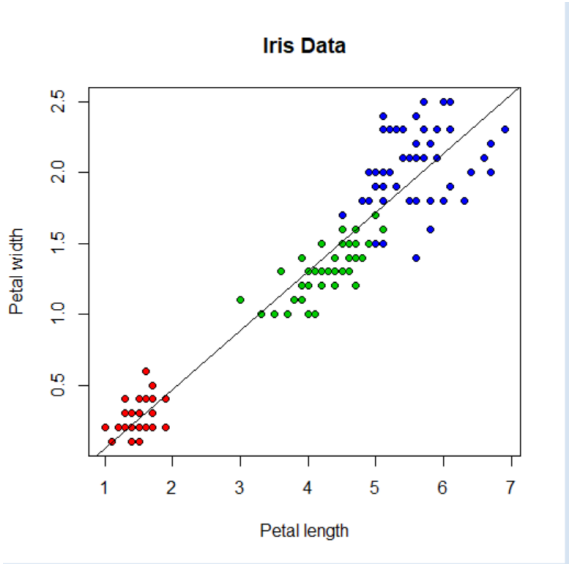
PRACTICAL NO. 6

Aim: Practical of Simple/Multiple Linear Regression

SIMPLE :

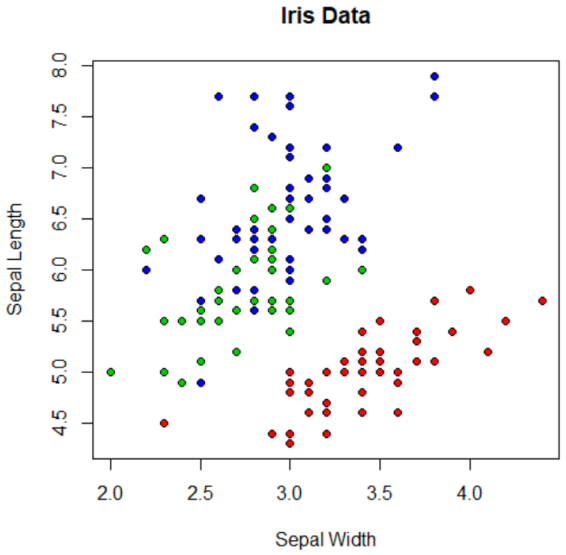
```
type  q()  to quit R.

> lsfit(iris$Petal.Length, iris$Petal.Width)$coefficients
 Intercept          X
-0.3630755  0.4157554
>
> plot(iris$Petal.Length, iris$Petal.Width, pch=21, bg=c("red","green3","blue")[uncla$
+ Species)], main="Iris Data", xlab="Petal length", ylab="Petal width")
>
> abline(lsfit(iris$Petal.Length, iris$Petal.Width)$coefficients, col="black")
>
```

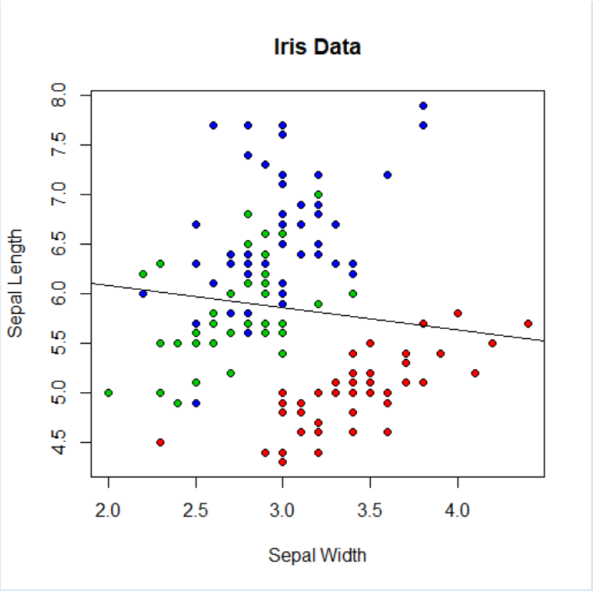





```
> #create smple regression model
> lm(Petal.Width ~ Petal.Length, data=iris)$coefficients
 (Intercept) Petal.Length
  -0.3630755    0.4157554
>
> plot(iris$Petal.Length, iris$Petal.Width, pch=21, bg=c("red","green3","blue")[uncla$
+ Species)], main="Iris Data", xlab="Petal length", ylab="Petal width")
>
> abline(lm(Petal.Width ~ Petal.Length, data=iris)$coefficients, col="black")
>
> summary(lm(Petal.Width ~ Petal.Length, data=iris))

Call:
lm(formula = Petal.Width ~ Petal.Length, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-0.56515 -0.12358 -0.01898  0.13288  0.64272

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.363076   0.039762  -9.131  4.7e-16 ***
Petal.Length  0.415755   0.009582  43.387  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2065 on 148 degrees of freedom
Multiple R-squared:  0.9271,    Adjusted R-squared:  0.9266
F-statistic:  1882 on 1 and 148 DF,  p-value: < 2.2e-16

>
```

```
> plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue")$
+ $Species)], main="Iris Data", xlab="Sepal Width", ylab="Sepal Length")
> |
```





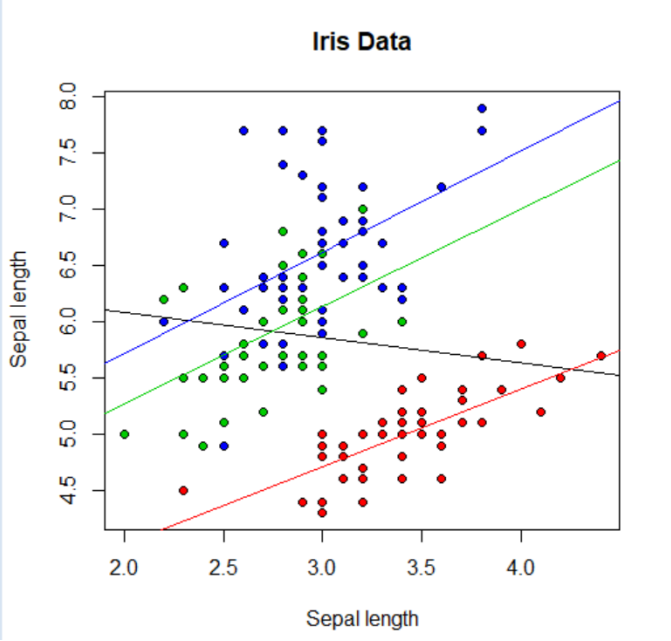

```
> plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue")$
+ $Species)], main="Iris Data", xlab="Sepal Width", ylab="Sepal Length")
> abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients, col="black")
> summary(lm(Sepal.Length ~ Sepal.Width, data=iris))

Call:
lm(formula = Sepal.Length ~ Sepal.Width, data = iris)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5561 -0.6333 -0.1120  0.5579  2.2226

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.5262     0.4789   13.63   <2e-16 ***
Sepal.Width   -0.2234     0.1551   -1.44    0.152
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8251 on 148 degrees of freedom
Multiple R-squared:  0.01382,   Adjusted R-squared:  0.007159
F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519

> |
```

Multiple:

#What happens if we divide the data up by species, and run three separate linear regressions?

```
> plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue"$
+ $Species)], main="Iris Data", xlab="Sepal Width", ylab="Sepal Length")
> abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients, col="black")
> summary(lm(Sepal.Length ~ Sepal.Width, data=iris))

Call:
lm(formula = Sepal.Length ~ Sepal.Width, data = iris)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5561 -0.6333 -0.1120  0.5579  2.2226

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.5262     0.4789   13.63   <2e-16 ***
Sepal.Width   -0.2234     0.1551   -1.44    0.152
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8251 on 148 degrees of freedom
Multiple R-squared:  0.01382,   Adjusted R-squared:  0.007159
F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

```
> plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue")$
+ $Species)], main="Iris Data", xlab="Sepal length", ylab="Sepal length")
> abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients, col="black")
> abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="setosa")$
> abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="versicol$
> abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="virginic$
>
>
```



```
> #The coefficients doing separate per species regressions of Sepal.Length ~ S$
> lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="setosa"),])$co$
(Intercept) Sepal.Width
  2.6390012   0.6904897
> lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="versicolor"),]$
(Intercept) Sepal.Width
  3.5397347   0.8650777
> lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="virginica"),])$
(Intercept) Sepal.Width
  3.9068365   0.9015345
> lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris)$coefficients
               Speciessetosa            Speciesversicolor
                   2.6390012                   3.5397347
             Speciesvirginica     Sepal.Width:Speciessetosa
                   3.9068365                   0.6904897
Sepal.Width:Speciesversicolor  Sepal.Width:Speciesvirginica
                   0.8650777                   0.9015345
> #Using the summary command on the linear model object gives:
>
```

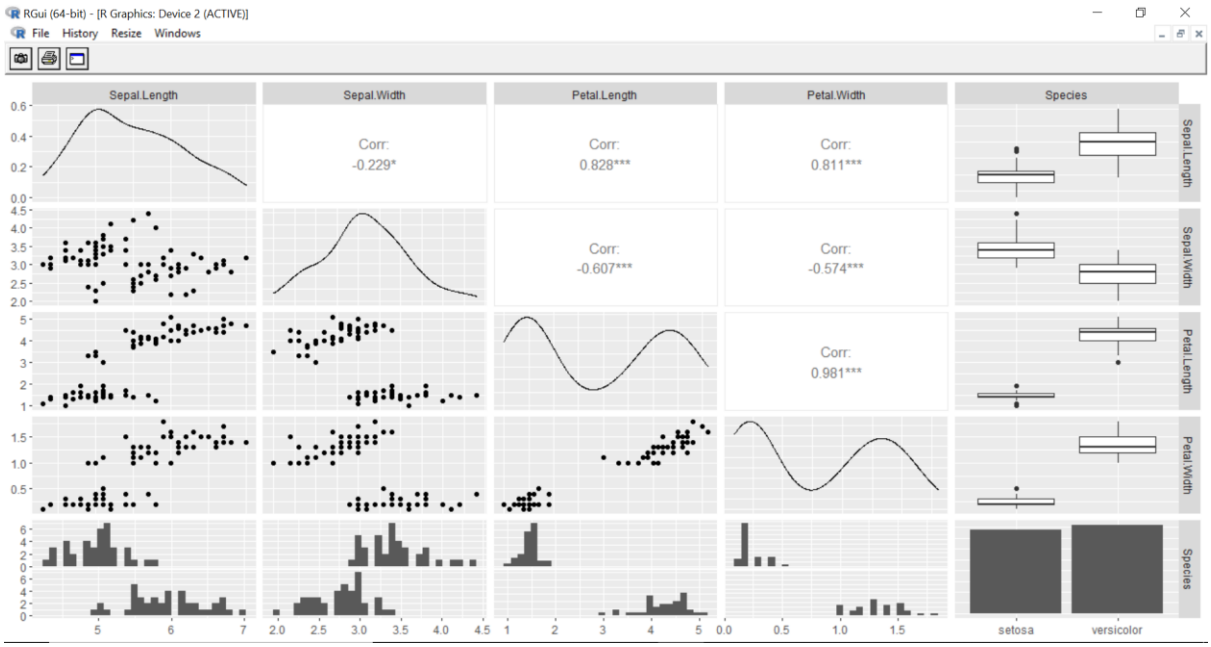
```
> summary(lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris))

Call:
lm(formula = Sepal.Length ~ Sepal.Width:Species + Species - 1,
    data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.26067 -0.25861 -0.03305  0.18929  1.44917

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
Speciessetosa                   2.6390     0.5715   4.618 8.53e-06 ***
Speciesversicolor               3.5397     0.5580   6.343 2.74e-09 ***
Speciesvirginica                3.9068     0.5827   6.705 4.25e-10 ***
Sepal.Width:Speciessetosa       0.6905     0.1657   4.166 5.31e-05 ***
Sepal.Width:Speciesversicolor   0.8651     0.2002   4.321 2.88e-05 ***
Sepal.Width:Speciesvirginica    0.9015     0.1948   4.628 8.16e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4397 on 144 degrees of freedom
Multiple R-squared:  0.9947,    Adjusted R-squared:  0.9944
F-statistic:  4478 on 6 and 144 DF,  p-value: < 2.2e-16
```

```
> summary(step(lm(Sepal.Length ~ Sepal.Width * Species, data=iris)))
Start:  AIC=-240.59
Sepal.Length ~ Sepal.Width * Species

                     Df Sum of Sq    RSS     AIC
- Sepal.Width:Species  2   0.15719 28.004 -243.75
<none>                             27.846 -240.59

Step:  AIC=-243.74
Sepal.Length ~ Sepal.Width + Species

             Df Sum of Sq     RSS     AIC
<none>                    28.004 -243.75
- Sepal.Width  1    10.953  38.956 -196.23
- Species      2    72.752 100.756  -55.69

Call:
lm(formula = Sepal.Length ~ Sepal.Width + Species, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.30711 -0.25713 -0.05325  0.19542  1.41253

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         2.2514     0.3698   6.089 9.57e-09 ***
Sepal.Width         0.8036     0.1063   7.557 4.19e-12 ***
Speciesversicolor   1.4587     0.1121  13.012  < 2e-16 ***
Speciesvirginica    1.9468     0.1000  19.465  < 2e-16 ***
---
>
> #I just introduced a model of the form Sepal.Length ~ Sepal.Width:Species + Species - 1,
> #which gave identical coefficients to those found doing species specific regressions:
> lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris)$coefficients
               Speciessetosa            Speciesversicolor
                   2.6390012                   3.5397347
             Speciesvirginica     Sepal.Width:Speciessetosa
                   3.9068365                   0.6904897
Sepal.Width:Speciesversicolor  Sepal.Width:Speciesvirginica
                   0.8650777                   0.9015345
> lm(Sepal.Length ~ Sepal.Width:Species + Species, data=iris)$coefficients
                 (Intercept)            Speciesversicolor
                   2.6390012                   0.9007335
             Speciesvirginica     Sepal.Width:Speciessetosa
                   1.2678352                   0.6904897
Sepal.Width:Speciesversicolor  Sepal.Width:Speciesvirginica
                   0.8650777                   0.9015345
>
```

PRACTICAL NO. 7

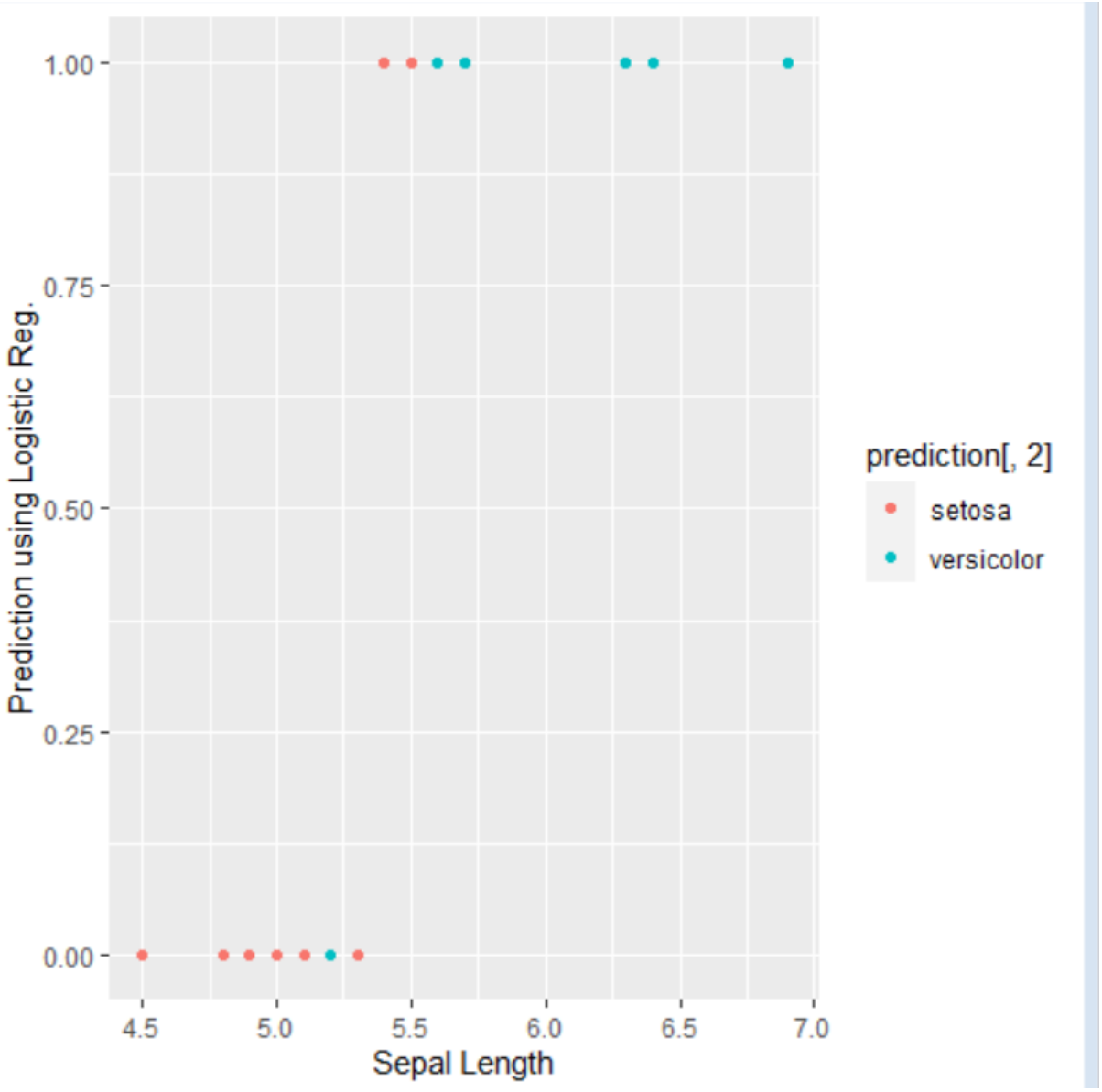
Aim: Practical of Logistics Regression

```
> 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$Species)
[1] "setosa"     "versicolor" "virginica"
> sum(is.na(ir_data))
[1] 0
> ir_data<-ir_data[1:100,]
> set.seed(100)
> samp<-sample(1:100,80)
> ir_test<-ir_data[samp,]
> ir_ctrl<-ir_data[-samp,]
> install.packages("ggplot2")
Installing package into 'C:/Users/nikhi/OneDrive/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cloud.r-project.org/bin/windows/contrib/4.1/ggplot2_3.3.5.z$
Content type 'application/zip' length 4129928 bytes (3.9 MB)
downloaded 3.9 MB

package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\nikhi\AppData\Local\Temp\Rtmp0k8V0p\downloaded_packages
> library(ggplot2)
Warning message:
package 'ggplot2' was built under R version 4.1.3
> install.packages("GGally")
Installing package into 'C:/Users/nikhi/OneDrive/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
also installing the dependencies 'hms', 'prettyunits', 'forcats', 'progress', '$
The downloaded binary packages are in
        C:\Users\nikhi\AppData\Local\Temp\Rtmp0k8V0p\downloaded_packages
> library(GGally)
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
Warning message:
package 'GGally' was built under R version 4.1.3
> ggpairs(ir_test)
 plot: [5,1] [=========================================>--------] 84% est: 0s `$
 plot: [5,2] [==========================================>------] 88% est: 0s `$
 plot: [5,3] [============================================>----] 92% est: 0s `$
 plot: [5,4] [=============================================>--] 96% est: 0s `$
>
```




```
> 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
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

#t.test(dataset,dataset,alternative method, mu value, var.equal=F, conf.level=0.95)

```
> x=c(418,421,421,422,425,427,431,434,437,439,446,447,448,453,454
+
+ ,463,465)
> y=c(429,430,430,431,436,437,440,441,445,446,447)
> 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 = 0.19937, df = 23.869, p-value = 0.8437
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.854361  9.533506
sample estimates:
mean of x mean of y
 438.2941  437.4545
```


```
> x=c(418,421,421,422,425,427,431,434,437,439,446,447,448,453,454
+
+ ,463,465)
> y=c(429,430,430,431,436,437,440,441,445,446,447)
> 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 = 0.19937, df = 23.869, p-value = 0.8437
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.854361  9.533506
sample estimates:
mean of x mean of y
 438.2941  437.4545

>
```

PRACTICAL NO. 9

Aim: Practical of Analysis of Variance

```
> y1 = c(18.2, 20.1, 17.6, 16.8, 18.8, 19.7, 19.1)
> y2 = c(17.4, 18.7, 19.1, 16.4, 15.9, 18.4, 17.7)
> y3 = c(15.2, 18.8, 17.7, 16.5, 15.9, 17.1, 16.7)
> y = c(y1, y2, y3)
> n = rep(7, 3)
> n
[1] 7 7 7
> group = rep(1:3, n)
> group
 [1] 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 3
> tmp = tapply(y, group, stem)

  The decimal point is at the |

  16 | 8
  17 | 6
  18 | 28
  19 | 17
  20 | 1


  The decimal point is at the |

  15 | 9
  16 | 4
  17 | 47
  18 | 47
  19 | 1



  The decimal point is at the |

  15 | 29
  16 | 57
  17 | 17
  18 | 8

> stem(y)

  The decimal point is at the |

  15 | 299
  16 | 4578
  17 | 14677
  18 | 24788
  19 | 117
  20 | 1

> tmpfn = function(x) c(sum = sum(x), mean = mean(x), var = var(x),n = length(x))
> tapply(y, group, tmpfn)
$`1`
       sum       mean        var          n
130.300000  18.614286   1.358095   7.000000

$`2`
       sum       mean        var          n
123.600000  17.657143   1.409524   7.000000

$`3`
       sum       mean        var          n
117.900000  16.842857   1.392857   7.000000

> tmpfn(y)
       sum       mean        var          n
371.800000  17.704762   1.798476  21.000000
```

```
> data = data.frame(y = y, group = factor(group))
> fit = lm(y ~ group, data)
> anova(fit)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
group      2 11.007  5.5033  3.9683 0.03735 *
Residuals 18 24.963  1.3868
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> df = anova(fit)[, "Df"]
> names(df) = c("trt", "err")
> df
trt err
  2  18
> alpha = c(0.05, 0.01)
> qf(alpha, df["trt"], df["err"], lower.tail = FALSE)
[1] 3.554557 6.012905
> anova(fit)["Residuals", "Sum Sq"]
[1] 24.96286
> anova(fit)["Residuals", "Sum Sq"]/qchisq(c(0.025, 0.975), 18,lower.tail = FALSE)
[1] 0.7918086 3.0328790
>
```

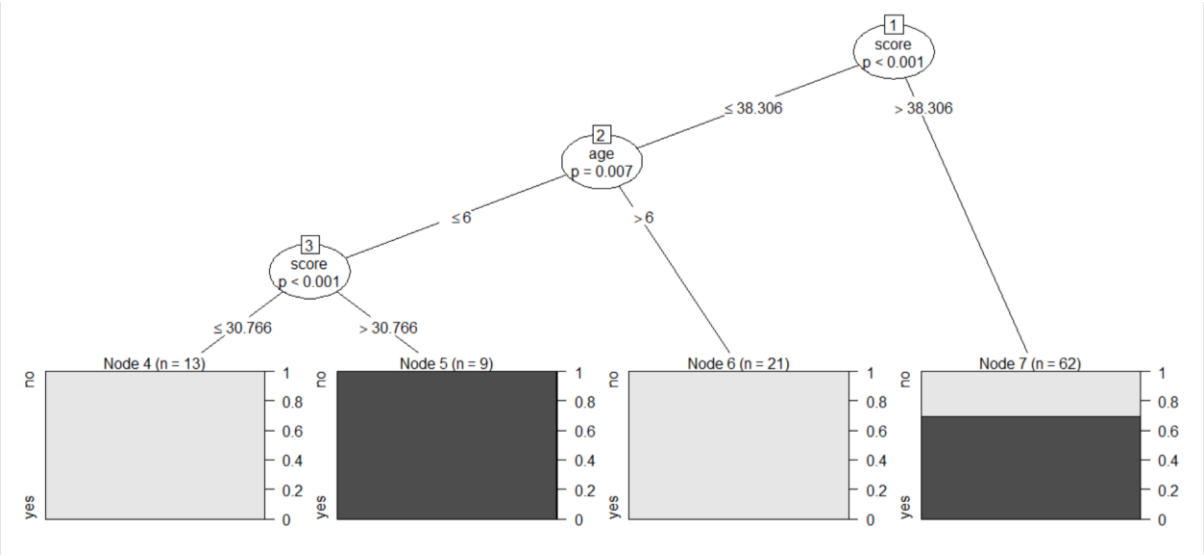
PRACTICAL NO. 10

Aim: Practical of Decision Tree

```
> install.packages('party')
Installing package into 'C:/Users/nikhi/OneDrive/Documents/R/win-library/4.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
also installing the dependencies 'TH.data', 'libcoin', 'matrixStats', 'multcomp$

trying URL 'https://cloud.r-project.org/bin/windows/contrib/4.1/TH.data_1.1-0.z$
Content type 'application/zip' length 8807478 bytes (8.4 MB)
downloaded 8.4 MB

trying URL 'https://cloud.r-project.org/bin/windows/contrib/4.1/libcoin_1.0-9.z$
```

```
> 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'
```

```
> 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
> input.dat<-readingSkills[c(1:105),]
> png(file="decision_tree.png")
> output.tree<-ctree(nativeSpeaker~age+shoeSize+score,data=input.dat)
> plot(output.tree)
> dev.off()
null device
          1
> plot(output.tree)
>
```



Conclusion: We conclude that anyone whose reading skill score is less than 38.3 and age is more than 6 is a native speaker