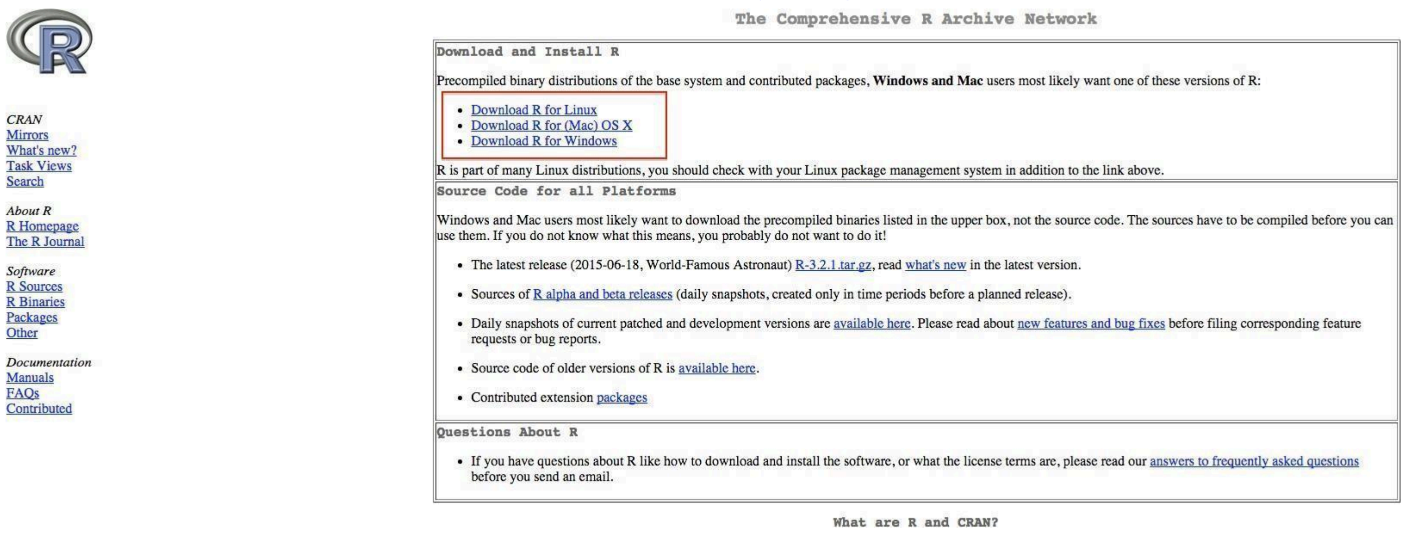
Roll.no: 210701277

EX. NO: 7

INSTALLATION GUIDE FOR R AND R STUDIO

Step 1 – Install R

1. Download the R installer from https://cran.r--‐project.org/


The Comprehensive R Archive Network

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-06-18, World-Famous Astronaut) R-3.2.1.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

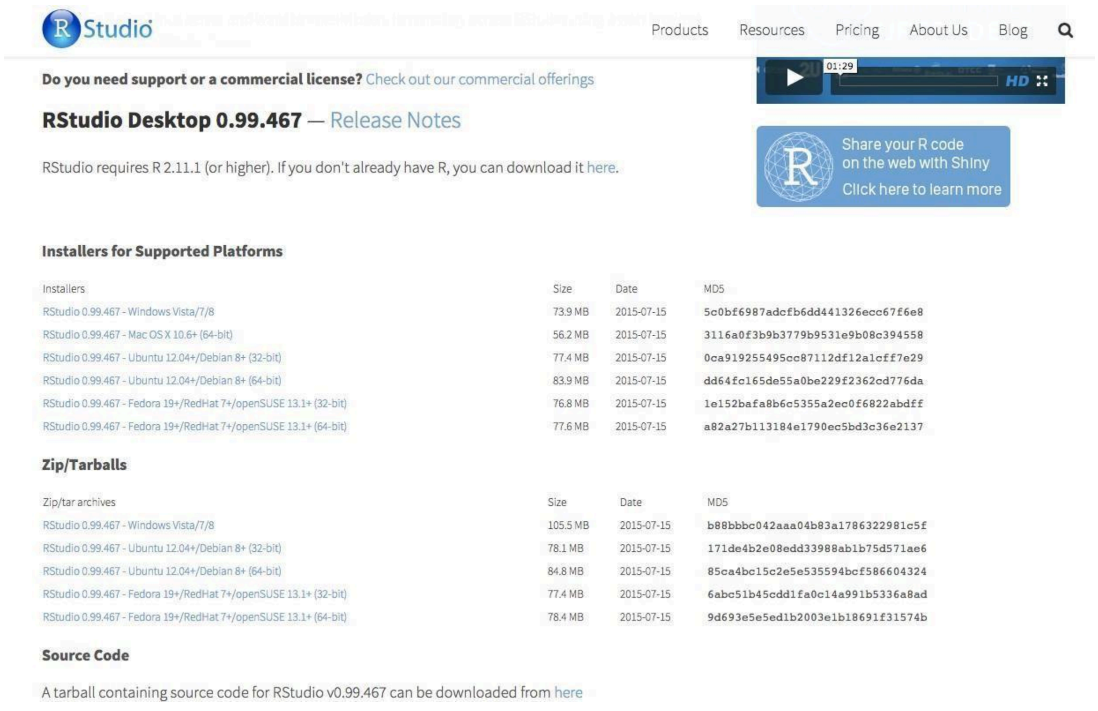
What are R and CRAN?

Figure 1. Screenshot of http://cran.csiro.au/

2. Run the installer. Default settings are fine. If you do not have admin rights on your laptop, then ask you local IT support. In that case, it is important that you also ask them to give you full permissions to the R directories. Without this, you will not be able to install additional packages later

Step 2 – Install RStudio

1. Download RStudio: https://www.rstudio.com/products/rstudio/download/



Do you need support or a commercial license? Check out our commercial offerings

RStudio Desktop 0.99.467 — Release Notes

RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it here.

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 0.99.467 - Windows Vista/7/8	73.9 MB	2015-07-15	5c0bf6987adcfb6dd441326ecc67f6e8
RStudio 0.99.467 - Mac OS X 10.6+ (64-bit)	56.2 MB	2015-07-15	3116a0f3b9b3779b9531e9b08c394558
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (32-bit)	77.4 MB	2015-07-15	0ca919255495cc87112df12a1cff7e29
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (64-bit)	83.9 MB	2015-07-15	dd64fc165de55a0be229f2362cd776da
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	76.8 MB	2015-07-15	1e152bafa8b6c5355a2ec0f6822abdff
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	77.6 MB	2015-07-15	a82a27b113184e1790ec5bd3c36e2137

Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 0.99.467 - Windows Vista/7/8	105.5 MB	2015-07-15	b88bbbc042aaa04b83a1786322981c5f
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (32-bit)	78.1 MB	2015-07-15	171de4b2e08edd33988ab1b75d571ae6
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (64-bit)	84.8 MB	2015-07-15	85ca4bc15c2e5e535594bcf586604324
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	77.4 MB	2015-07-15	6abc51b45cdd1fa0c14a991b5336a8ad
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	78.4 MB	2015-07-15	9d693e5e5ed1b2003e1b18691f31574b

Source Code

A tarball containing source code for RStudio v0.99.467 can be downloaded from here

Figure 2. Download RStudio on https://www.rstudio.com/products/rstudio/download/

2. Once the installation of R has completed successfully (and not before), run the RStudio installer.
3. If you do not have administrative rights on your laptop, step 2 may fail. Ask your IT Support or download a pre--‑built zip archive of RStudio which doesn't need installing. The link for this is towards the bottom of the download page, highlighted in Image 2.

   a. Download the appropriate archive for your system (Windows/Linux only – the Mac version can be installed into your personal "Applications" folder without admin rights).

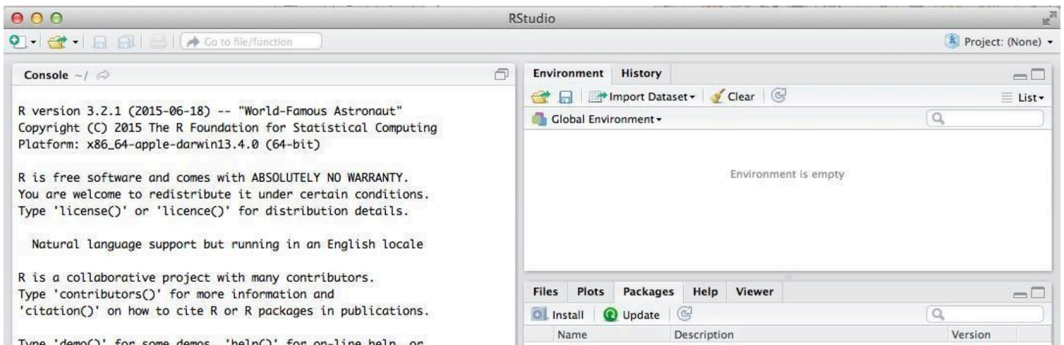
   b. Double clicking on the zip archive should automatically unpack it on most Windows machines.

Step 3 – Check that R and RStudio are working

1. Open RStudio. It should open a window that looks similar to image 3 below.

2. In the left hand window, by the '>'sign, type '4+5'(without the quotes) and hit enter. An output line reading '[1] 9' should appear. This means that R and RStudio are working.

3. If this is not successful, contact us or your local IT support for further advice

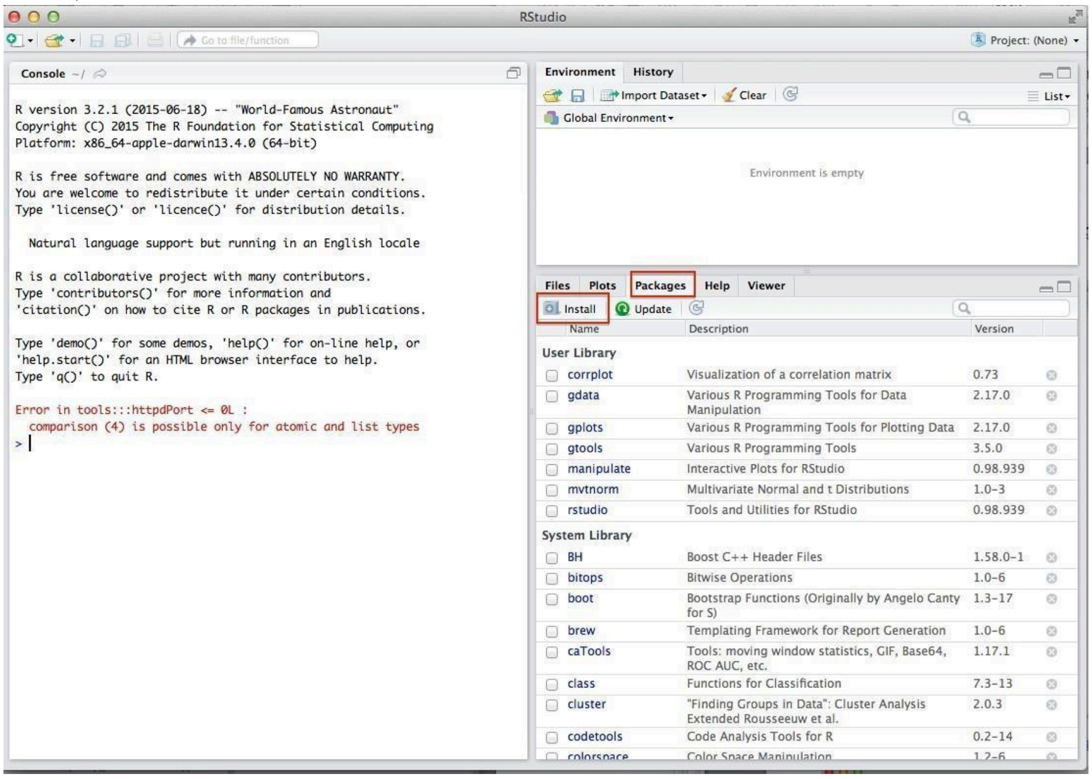


```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

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.

  Natural language support but running in an English locale

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
```

Figure 3. Running R with RStudio

Step 4 – Install R packages required for the workshop

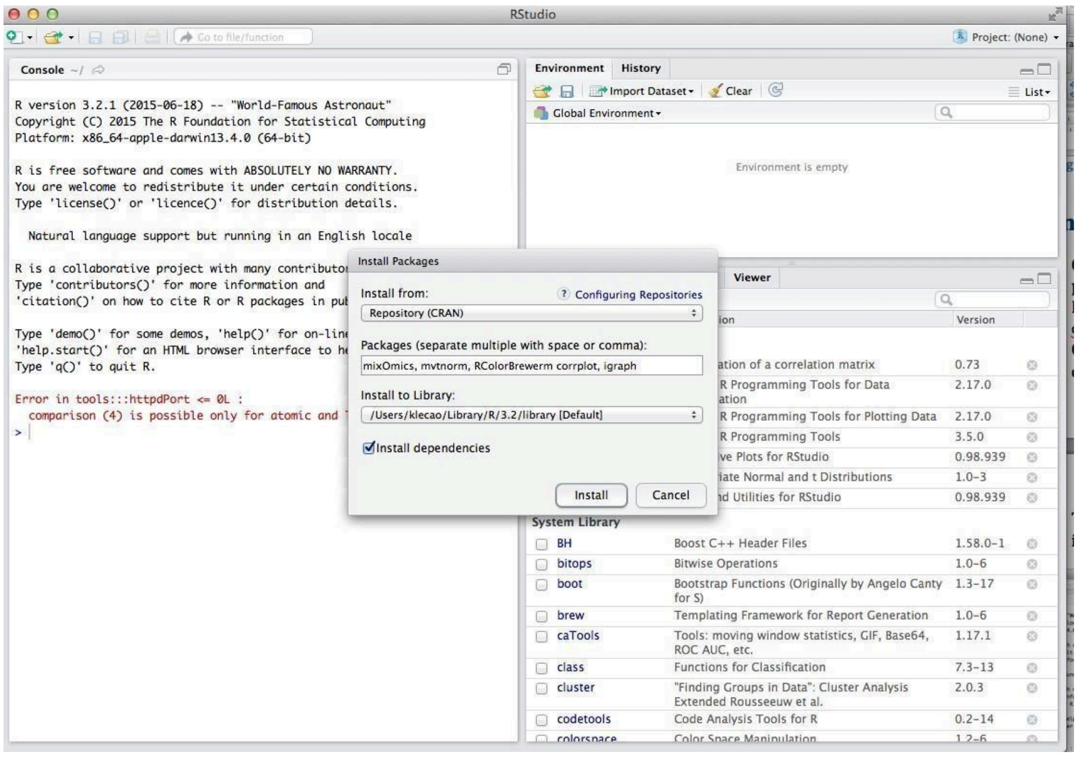
1. Click on the tab ‘ Packages’ then ‘Install’ as shown in Image 4. Or Tools --–> Install packages.
2. Install the following packages: mixOmics version 6.1.0, mvtnorm, RColorBrewer, corrplot, igraph (see Image 4). For apple mac users, if you are unable to install the mixOmics imported library rgl, you will need to install the XQuartz software first https://www.xquartz.org/
3. Check that the packages are installed by typing ‘library(mixOmics)’ (without the quotes) in the prompt and press enter (see Image 5).
4. Then type ‘sessionInfo()’ and check that mixOmics version 6.1.0 has been installed (image 6).



```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (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.

  Natural language support but running in an English locale

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.

Error in tools:::httpdPort <= 0L :
  comparison (4) is possible only for atomic and list types
>
```

Name	Description	Version
User Library		
corrplot	Visualization of a correlation matrix	0.73
gdata	Various R Programming Tools for Data Manipulation	2.17.0
gplots	Various R Programming Tools for Plotting Data	2.17.0
gtools	Various R Programming Tools	3.5.0
manipulate	Interactive Plots for RStudio	0.98.939
mvtnorm	Multivariate Normal and t Distributions	1.0-3
rstudio	Tools and Utilities for RStudio	0.98.939
System Library		
BH	Boost C++ Header Files	1.58.0-1
bitops	Bitwise Operations	1.0-6
boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-17
brew	Templating Framework for Report Generation	1.0-6
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
class	Functions for Classification	7.3-13
cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.3
codetools	Code Analysis Tools for R	0.2-14
colorspace	Color Space Manipulation	1.2-6

Figure 4. Click on Install to install R packages.



```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (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.

  Natural language support but running in an English locale

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

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

Error in tools:::httpdPort <= 0L :
  comparison (4) is possible only for atomic and
>
```

Install Packages

Install from: Repository (CRAN) ? Configuring Repositories

Packages (separate multiple with space or comma): mixOmics, mvtnorm, RColorBrewerm corrplot, igraph

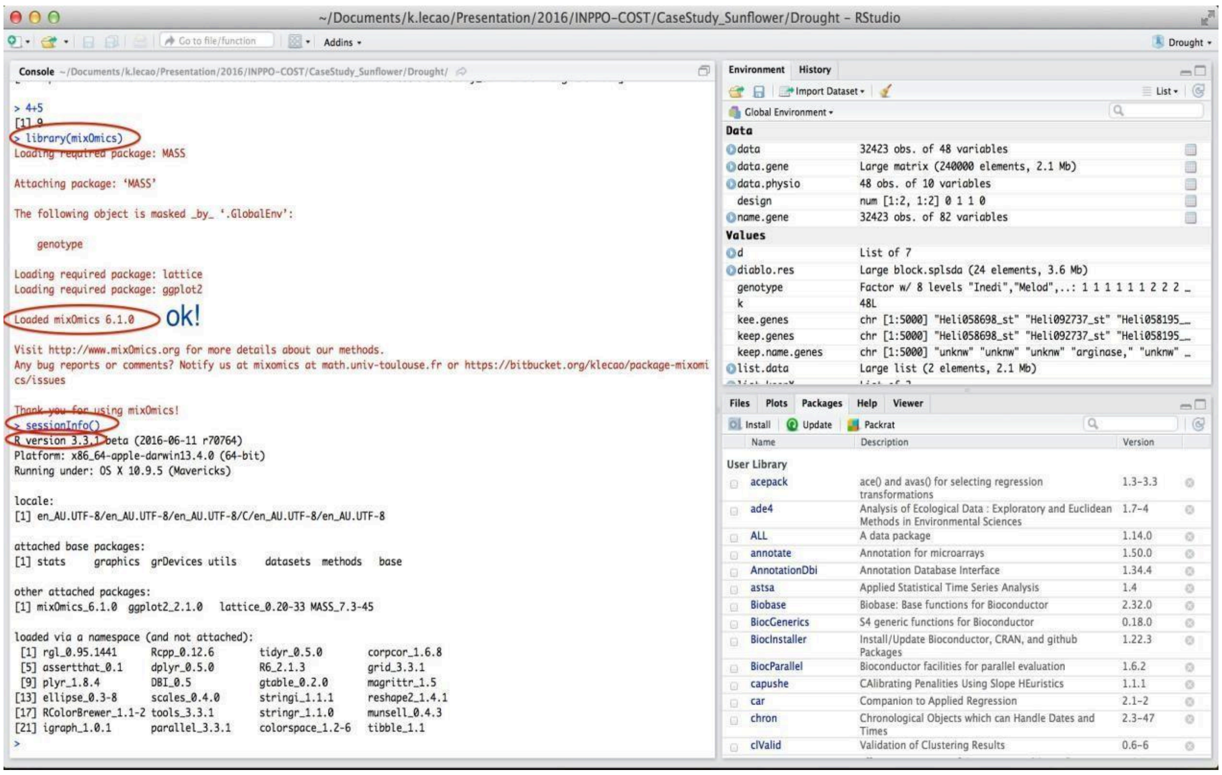
Install to Library: /Users/klecao/Library/R/3.2/library [Default]

☑ Install dependencies

Install Cancel

		ion	Version
		ation of a correlation matrix	0.73
		R Programming Tools for Data ation	2.17.0
		R Programming Tools for Plotting Data	2.17.0
		R Programming Tools	3.5.0
		ve Plots for RStudio	0.98.939
		iate Normal and t Distributions	1.0-3
		nd Utilities for RStudio	0.98.939
System Library			
☐	BH	Boost C++ Header Files	1.58.0-1
☐	bitops	Bitwise Operations	1.0-6
☐	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-17
☐	brew	Templating Framework for Report Generation	1.0-6
☐	caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
☐	class	Functions for Classification	7.3-13
☐	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.3
☐	codetools	Code Analysis Tools for R	0.2-14
☐	colorspace	Color Space Manipulation	1.2-6

Figure 5. Specify the list of packages to be installed


~/Documents/k.lecao/Presentation/2016/INPPO-COST/CaseStudy_Sunflower/Drought – RStudio
Go to file/function
Addins
Drought
Console ~/Documents/k.lecao/Presentation/2016/INPPO-COST/CaseStudy_Sunflower/Drought/
ok!
Environment
History
Import Dataset
List
Global Environment
Files
Plots
Packages
Help
Viewer
Install
Update
Packrat

```
> 4+5
[1] 9
> library(mixOmics)
Loading required package: MASS

Attaching package: 'MASS'

The following object is masked _by_ '.GlobalEnv':

    genotype

Loading required package: lattice
Loading required package: ggplot2

Loaded mixOmics 6.1.0

Visit http://www.mixOmics.org for more details about our methods.
Any bug reports or comments? Notify us at mixomics at math.univ-toulouse.fr or https://bitbucket.org/klecao/package-mixomi
cs/issues

Thank you for using mixOmics!
> sessionInfo()
R version 3.3.1 beta (2016-06-11 r70764)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
[1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] mixOmics_6.1.0  ggplot2_2.1.0   lattice_0.20-33 MASS_7.3-45

loaded via a namespace (and not attached):
 [1] rgl_0.95.1441      Rcpp_0.12.6        tidyr_0.5.0        corpcor_1.6.8
 [5] assertthat_0.1     dplyr_0.5.0        R6_2.1.3           grid_3.3.1
 [9] plyr_1.8.4         DBI_0.5            gtable_0.2.0       magrittr_1.5
[13] ellipse_0.3-8      scales_0.4.0       stringi_1.1.1      reshape2_1.4.1
[17] RColorBrewer_1.1-2 tools_3.3.1        stringr_1.1.0      munsell_0.4.3
[21] igraph_1.0.1       parallel_3.3.1     colorspace_1.2-6   tibble_1.1
>
```

```
Data
data             32423 obs. of 48 variables
data.gene        Large matrix (240000 elements, 2.1 Mb)
data.physio      48 obs. of 10 variables
design           num [1:2, 1:2] 0 1 1 0
name.gene        32423 obs. of 82 variables
Values
d                List of 7
diablo.res       Large block.splsda (24 elements, 3.6 Mb)
genotype         Factor w/ 8 levels "Inedi","Melod",..: 1 1 1 1 1 1 2 2 2 _
k                48L
kee.genes        chr [1:5000] "Heli058698_st" "Heli092737_st" "Heli058195__
keep.genes       chr [1:5000] "Heli058698_st" "Heli092737_st" "Heli058195__
keep.name.genes  chr [1:5000] "unknw" "unknw" "unknw" "arginase," "unknw" _
list.data        Large list (2 elements, 2.1 Mb)
```

Name	Description	Version
User Library		
acepack	ace() and avas() for selecting regression transformations	1.3-3.3
ade4	Analysis of Ecological Data : Exploratory and Euclidean Methods in Environmental Sciences	1.7-4
ALL	A data package	1.14.0
annotate	Annotation for microarrays	1.50.0
AnnotationDbi	Annotation Database Interface	1.34.4
astsa	Applied Statistical Time Series Analysis	1.4
Biobase	Biobase: Base functions for Bioconductor	2.32.0
BiocGenerics	S4 generic functions for Bioconductor	0.18.0
BiocInstaller	Install/Update Bioconductor, CRAN, and github Packages	1.22.3
BiocParallel	Bioconductor facilities for parallel evaluation	1.6.2
capushe	CAlibrating Penalities Using Slope HEuristics	1.1.1
car	Companion to Applied Regression	2.1-2
chron	Chronological Objects which can Handle Dates and Times	2.3-47
clValid	Validation of Clustering Results	0.6-6

Figure 6. Check that the package mixOmics is installed and has the version 6.1.0.

RESULT: Thus to install R and R Studio is successfully completed.

EX .NO:7a IMPLEMENT LINEAR AND LOGISTIC REGRESSION

AIM :

To implement Linear and Logistic regression using R language.

a)Linear regression

Sample data
heights <- c(150, 160, 165, 170, 175, 180, 185)
weights <- c(55, 60, 62, 68, 70, 75, 80)

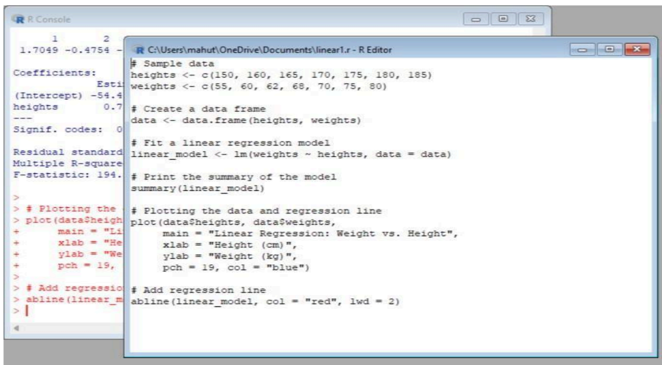
Create a data frame data <-
data.frame(heights, weights)

Fit a linear regression model linear_model <-
lm(weights ~ heights, data = data)

Print the summary of the model
print(summary(linear_model))

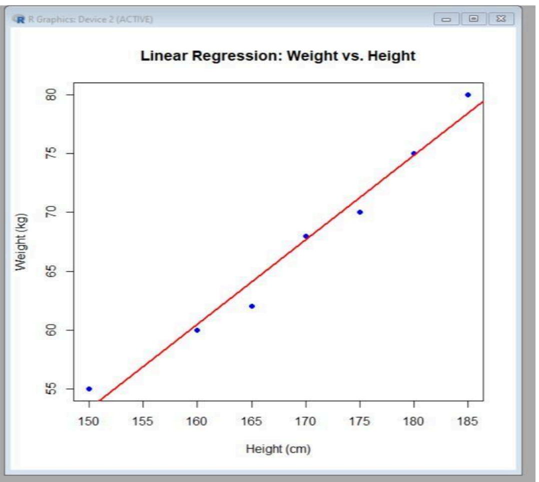
Plotting the data and regression line
plot(data$heights, data$weights,
main = "Linear Regression: Weight vs.
Height", xlab = "Height (cm)", ylab =
"Weight (kg)",
pch = 19, col = "blue")

Add regression line
abline(linear_model, col = "red", lwd = 2)



```
R Console
      1        2
 1.7049 -0.4754 -

Coefficients:
             Esti
(Intercept) -54.4
heights       0.7
---
Signif. codes:  0

Residual standard
Multiple R-square
F-statistic: 194.

>
> # Plotting the
> plot(data$heigh
+      main = "Li
+      xlab = "He
+      ylab = "We
+      pch = 19,
>
> # Add regressio
> abline(linear_m
>
```

```
C:\Users\mahut\OneDrive\Documents\linear1.r - R Editor
# Sample data
heights <- c(150, 160, 165, 170, 175, 180, 185)
weights <- c(55, 60, 62, 68, 70, 75, 80)

# Create a data frame
data <- data.frame(heights, weights)

# Fit a linear regression model
linear_model <- lm(weights ~ heights, data = data)

# Print the summary of the model
summary(linear_model)

# Plotting the data and regression line
plot(data$heights, data$weights,
     main = "Linear Regression: Weight vs. Height",
     xlab = "Height (cm)",
     ylab = "Weight (kg)",
     pch = 19, col = "blue")

# Add regression line
abline(linear_model, col = "red", lwd = 2)
```



b) Logistic regression

Load the dataset
data(mtcars)

Convert 'am' to a factor (categorical variable) mtcars$am <- factor(mtcars$am, levels = c(0, 1), labels = c("Automatic", "Manual"))

Fit a logistic regression model logistic_model <- glm(am ~ mpg, data = mtcars, family = binomial)

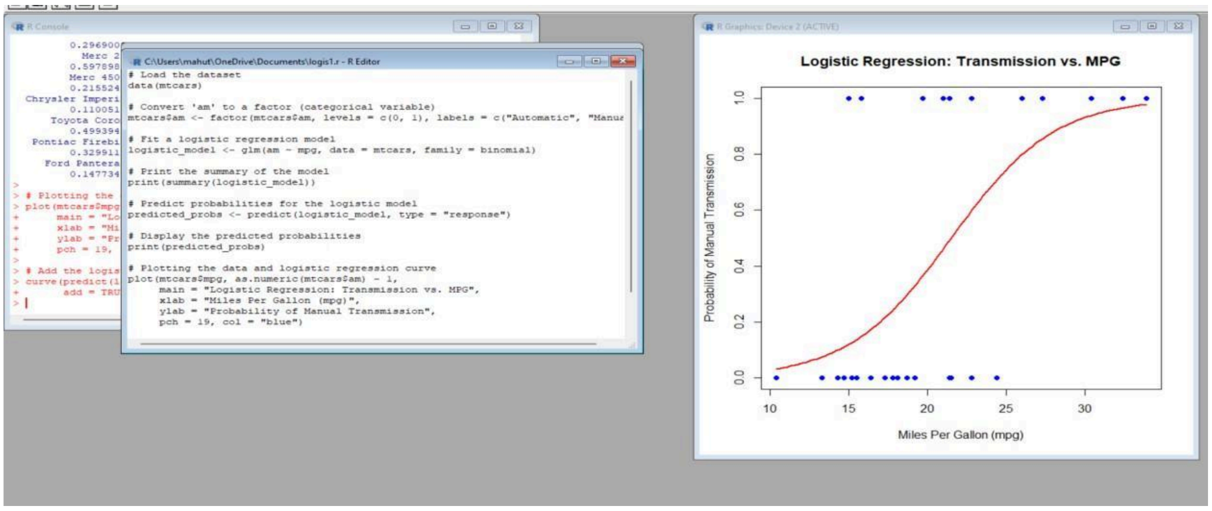
Print the summary of the model
print(summary(logistic_model))

Predict probabilities for the logistic model predicted_probs <- predict(logistic_model, type = "response")

Display the predicted probabilities print(predicted_probs)

Plotting the data and logistic regression curve plot(mtcars$mpg, as.numeric(mtcars$am) - 1, main = "Logistic Regression: Transmission vs. MPG", xlab = "Miles Per Gallon (mpg)", ylab = "Probability of Manual Transmission", pch = 19, col = "blue")

# Add the logistic regression curve
curve(predict(logistic_model, data.frame(mpg = x), type = "response"), add = TRUE, col = "red", lwd = 2)


R Console
C:\Users\mahut\OneDrive\Documents\logis1.r - R Editor
R Graphics: Device 2 (ACTIVE)
Logistic Regression: Transmission vs. MPG
Probability of Manual Transmission
Miles Per Gallon (mpg)
0.0
0.2
0.4
0.6
0.8
1.0
10
15
20
25
30

```
       0.296900
         Merc 2
       0.597898
       Merc 450
       0.215524
 Chrysler Imperi
       0.110051
    Toyota Coro
       0.499394
  Pontiac Firebi
       0.329911
    Ford Pantera
       0.147734
>
> # Plotting the
> plot(mtcars$mpg
+      main = "Lo
+      xlab = "Mi
+      ylab = "Pr
+      pch = 19,
>
> # Add the logis
> curve(predict(l
+        add = TRU
> |
```

```
# Load the dataset
data(mtcars)

# Convert 'am' to a factor (categorical variable)
mtcars$am <- factor(mtcars$am, levels = c(0, 1), labels = c("Automatic", "Manua

# Fit a logistic regression model
logistic_model <- glm(am ~ mpg, data = mtcars, family = binomial)

# Print the summary of the model
print(summary(logistic_model))

# Predict probabilities for the logistic model
predicted_probs <- predict(logistic_model, type = "response")

# Display the predicted probabilities
print(predicted_probs)

# Plotting the data and logistic regression curve
plot(mtcars$mpg, as.numeric(mtcars$am) - 1,
     main = "Logistic Regression: Transmission vs. MPG",
     xlab = "Miles Per Gallon (mpg)",
     ylab = "Probability of Manual Transmission",
     pch = 19, col = "blue")
```

RESULT:_ Thus to implement linear and logistic regression using R language is sucessfully done.

Ex.No:8

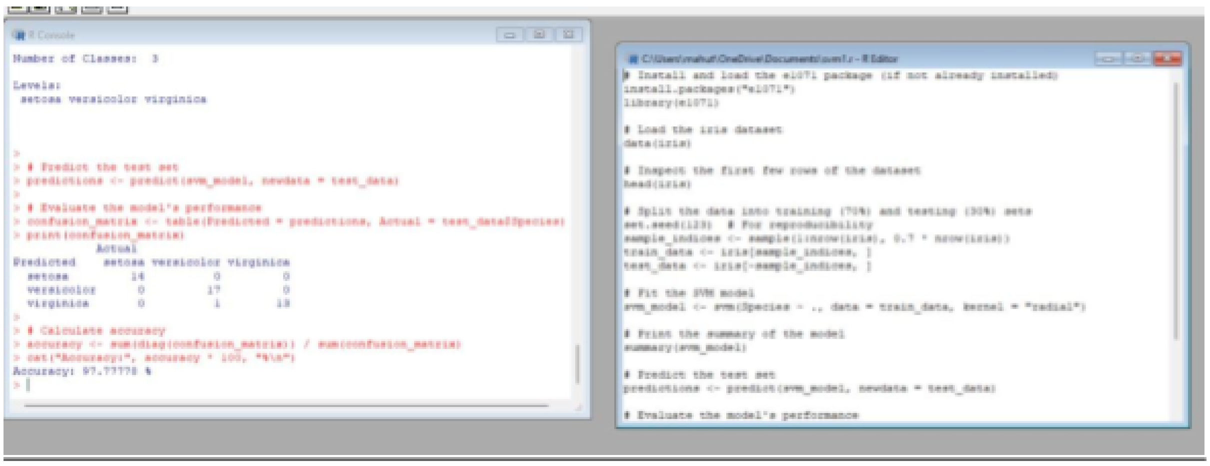
Implement SVM/Decision tree classification techniques

AIM :

to implement SVM and Decision tree classification techniques using R language.

a) SVM IN R

```
   # Install and load the e1071 package (if not already
installed) install.packages("e1071") library(e1071)

   # Load the iris dataset
data(iris)

   # Inspect the first few rows of the dataset
head(iris)

   # Split the data into training (70%) and testing (30%) sets
set.seed(123)  # For reproducibility
   sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[sample_indices, ]
   test_data <- iris[-sample_indices, ]

   # Fit the SVM model svm_model <- svm(Species ~ ., data =
train_data, kernel = "radial")

   # Print the summary of the model
   summary(svm_model)

   # Predict the test set predictions <-
predict(svm_model, newdata = test_data)

   # Evaluate the model's performance
   confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
print(confusion_matrix)

   # Calculate accuracy accuracy <- sum(diag(confusion_matrix))
/ sum(confusion_matrix) cat("Accuracy:", accuracy * 100, "%\n")
```



```
Number of Classes:  3

Levels:
 setosa versicolor virginica

>
> # Predict the test set
> predictions <- predict(svm_model, newdata = test_data)
>
> # Evaluate the model's performance
> confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
> print(confusion_matrix)
            Actual
Predicted    setosa versicolor virginica
  setosa         14          0         0
  versicolor      0         17         0
  virginica       0          1        13
>
> # Calculate accuracy
> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> cat("Accuracy:", accuracy * 100, "%\n")
Accuracy: 97.77778 %
>
```

```
# Install and load the e1071 package (if not already installed)
install.packages("e1071")
library(e1071)

# Load the iris dataset
data(iris)

# Inspect the first few rows of the dataset
head(iris)

# Split the data into training (70%) and testing (30%) sets
set.seed(123)  # For reproducibility
sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[sample_indices, ]
test_data <- iris[-sample_indices, ]

# Fit the SVM model
svm_model <- svm(Species ~ ., data = train_data, kernel = "radial")

# Print the summary of the model
summary(svm_model)

# Predict the test set
predictions <- predict(svm_model, newdata = test_data)

# Evaluate the model's performance
```

<u>**b) Decision tree in R**</u>

Install and load the rpart package (if not already installed) install.packages("rpart") library(rpart)

Load the iris dataset
data(iris)

Split the data into training (70%) and testing (30%) sets
set.seed(123) # For reproducibility
sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[sample_indices,]
test_data <- iris[-sample_indices,]

# Fit the Decision Tree model tree_model <- rpart(Species ~ ., data = train_data, method = "class")

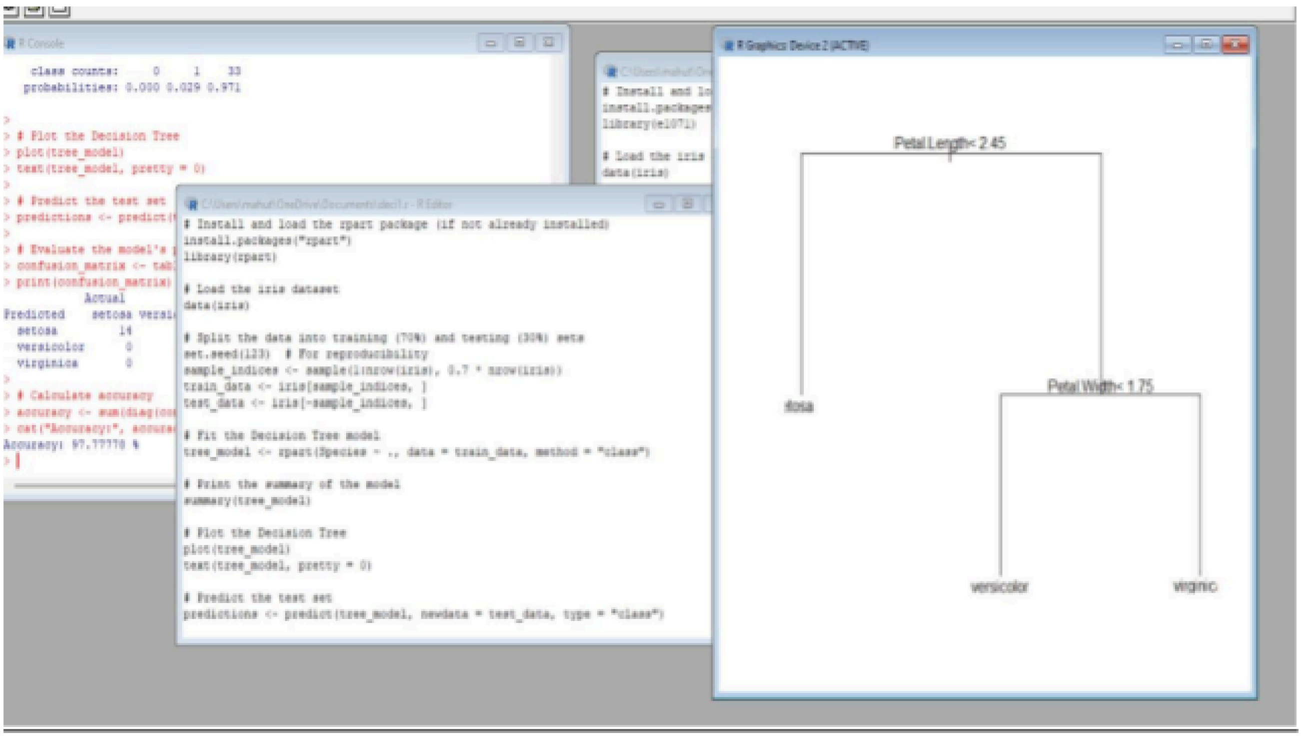
Print the summary of the model
summary(tree_model)

# Plot the Decision Tree
plot(tree_model) text(tree_model, pretty = 0)

Predict the test set predictions <- predict(tree_model, newdata = test_data, type = "class")

Evaluate the model's performance

```
    confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
print(confusion_matrix)

    # Calculate accuracy
    accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Accuracy:", accuracy * 100, "%\n")
```


R Console
C:\Users\mahut\OneDrive\Documents\dec1.r - R Editor
R Graphics Device 2 (ACTIVE)
Petal.Length< 2.45
Petal.Width< 1.75
tosa
versicolor
virginic

```
  class counts:     0     1    33
 probabilities: 0.000 0.029 0.971

>
> # Plot the Decision Tree
> plot(tree_model)
> text(tree_model, pretty = 0)
>
> # Predict the test set
> predictions <- predict(
>
> # Evaluate the model's p
> confusion_matrix <- tab
> print(confusion_matrix)
            Actual
Predicted    setosa versi
  setosa         14
  versicolor      0
  virginica       0
>
> # Calculate accuracy
> accuracy <- sum(diag(con
> cat("Accuracy:", accura
Accuracy: 97.77778 %
>
```

```
# Install and load the rpart package (if not already installed)
install.packages("rpart")
library(rpart)

# Load the iris dataset
data(iris)

# Split the data into training (70%) and testing (30%) sets
set.seed(123)  # For reproducibility
sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[sample_indices, ]
test_data <- iris[-sample_indices, ]

# Fit the Decision Tree model
tree_model <- rpart(Species ~ ., data = train_data, method = "class")

# Print the summary of the model
summary(tree_model)

# Plot the Decision Tree
plot(tree_model)
text(tree_model, pretty = 0)

# Predict the test set
predictions <- predict(tree_model, newdata = test_data, type = "class")
```

```
# Install and lo
install.package
library(e1071)

# Load the iris
data(iris)
```

RESULT :

Thus to implement SVM and Decision tree using R language is successfully completed.

Ex.No:9

Implement clustering techniques – Hierarchical and K-Means

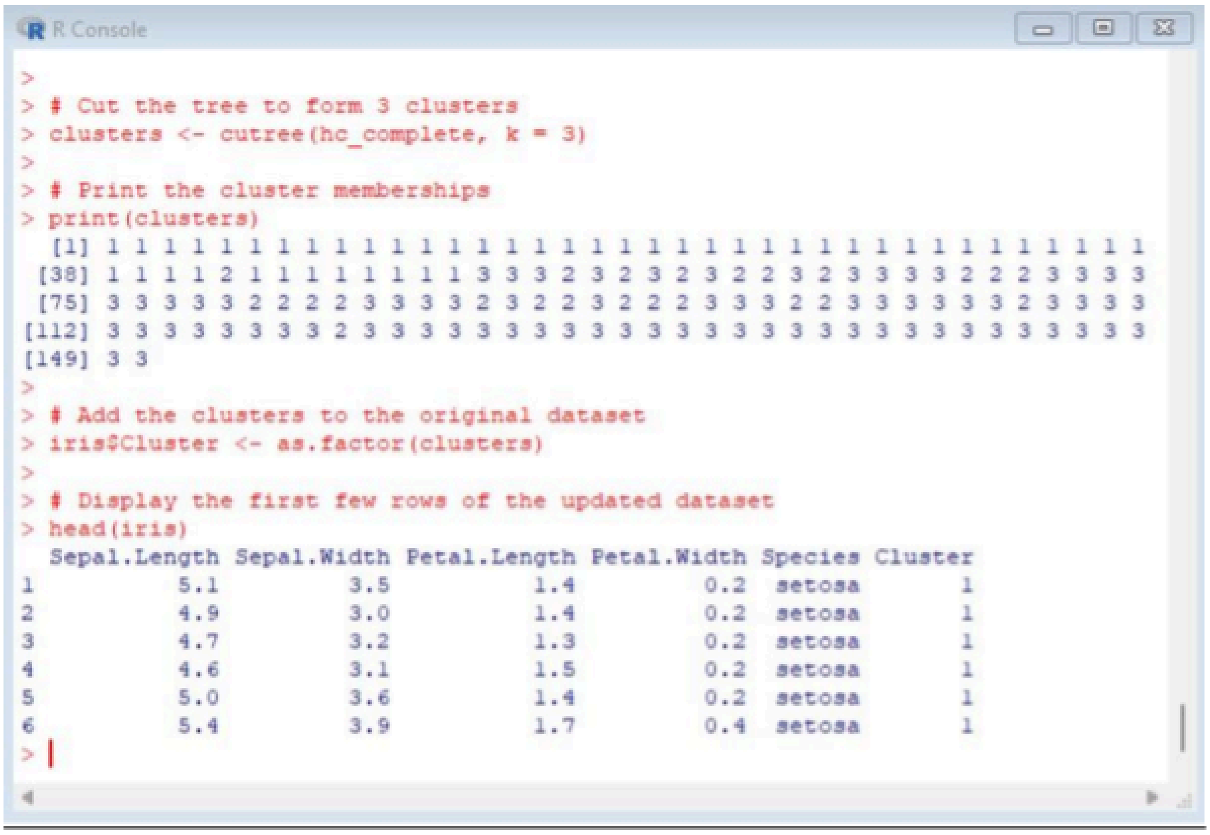
AIM:

To implement clustering techniques (hierarchical and K Means) using R language.

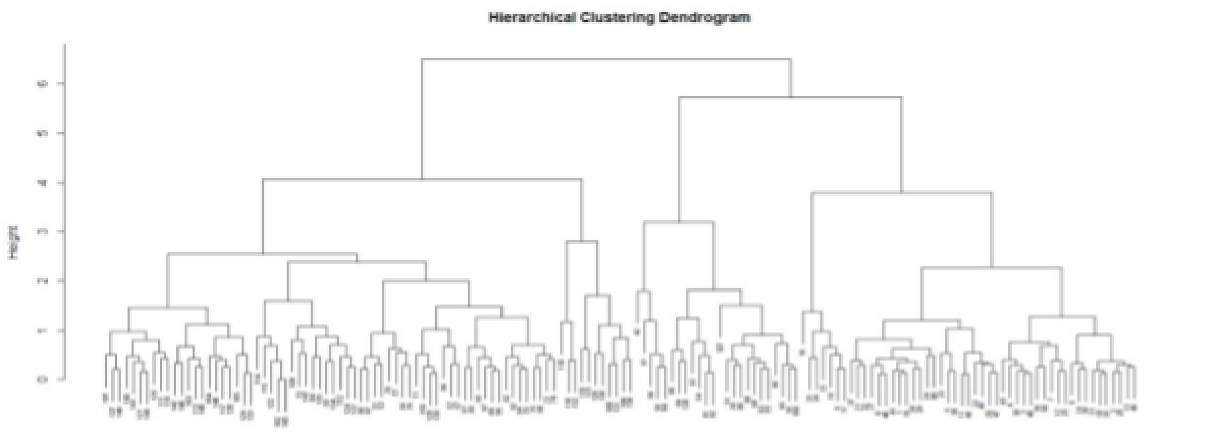
a) HIERARCHIAL CLUSTERING

```
   # Load the iris dataset
data(iris)

   # Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]

   # Standardize the data
   iris_scaled <- scale(iris_data)

   # Compute the distance matrix distance_matrix <-
dist(iris_scaled, method = "euclidean")

   # Perform hierarchical clustering using the "complete" linkage method
hc_complete <- hclust(distance_matrix, method = "complete")

   # Plot the dendrogram plot(hc_complete, main = "Hierarchical Clustering Dendrogram",
xlab = "", sub = "", cex =
   0.6)

   # Cut the tree to form 3 clusters
   clusters <- cutree(hc_complete, k = 3)

   # Print the cluster memberships
print(clusters)

   # Add the clusters to the original dataset
   iris$Cluster <- as.factor(clusters)
```

Display the first few rows of the updated dataset
head(iris)

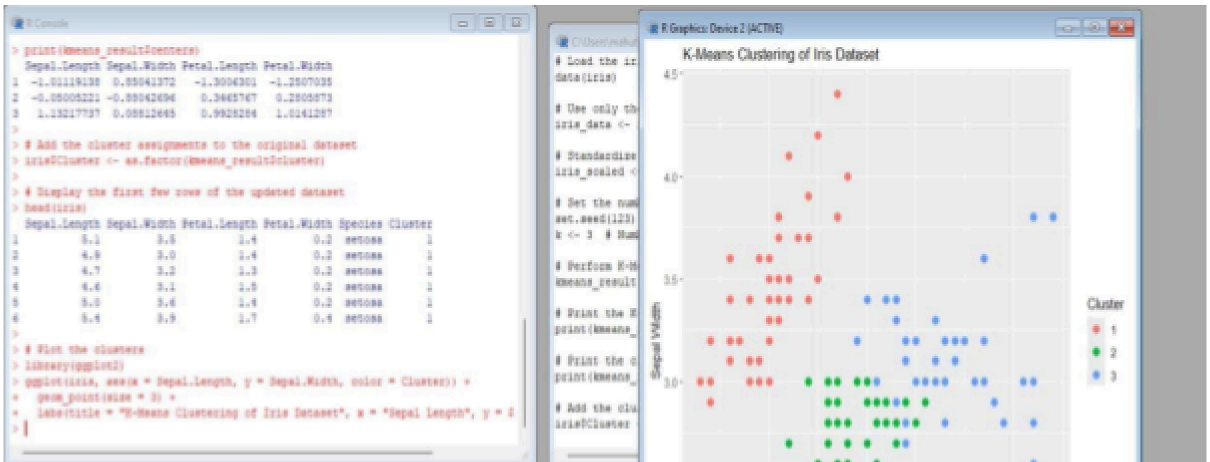


```
R Console
>
> # Cut the tree to form 3 clusters
> clusters <- cutree(hc_complete, k = 3)
>
> # Print the cluster memberships
> print(clusters)
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [38] 1 1 1 1 2 1 1 1 1 1 1 1 1 3 3 3 2 3 2 3 2 3 2 2 3 2 3 3 3 3 2 2 2 3 3 3 3
 [75] 3 3 3 3 3 2 2 2 2 3 3 3 3 2 3 2 2 3 2 2 2 3 3 3 2 2 3 3 3 3 3 3 2 3 3 3 3
[112] 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[149] 3 3
>
> # Add the clusters to the original dataset
> iris$Cluster <- as.factor(clusters)
>
> # Display the first few rows of the updated dataset
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species Cluster
1          5.1         3.5          1.4         0.2  setosa       1
2          4.9         3.0          1.4         0.2  setosa       1
3          4.7         3.2          1.3         0.2  setosa       1
4          4.6         3.1          1.5         0.2  setosa       1
5          5.0         3.6          1.4         0.2  setosa       1
6          5.4         3.9          1.7         0.4  setosa       1
> |
```



b) K-MEANS CLUSTERING

```
    # Load the iris dataset
data(iris)

    # Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]

    # Standardize the data
    iris_scaled <- scale(iris_data)

    # Set the number of clusters
set.seed(123)  # For reproducibility
    k <- 3  # Number of clusters

    # Perform K-Means clustering
    kmeans_result <- kmeans(iris_scaled, centers = k, nstart = 25)

    # Print the K-Means result
    print(kmeans_result)

    # Print the cluster centers
    print(kmeans_result$centers)

    # Add the cluster assignments to the original dataset
iris$Cluster <- as.factor(kmeans_result$cluster)

    # Display the first few rows of the updated dataset
head(iris)

    # Plot the clusters
library(ggplot2)
    ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster)) +
geom_point(size = 3) +
      labs(title = "K-Means Clustering of Iris Dataset", x = "Sepal Length", y = "Sepal Width")
```



```
> print(kmeans_result$centers)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -1.01119138  0.85041372   -1.3006301  -1.2507035
2  -0.05005221 -0.88042696    0.3465767   0.2805873
3   1.13217737  0.08812645    0.9928284   1.0141287
>
> # Add the cluster assignments to the original dataset
> iris$Cluster <- as.factor(kmeans_result$cluster)
>
> # Display the first few rows of the updated dataset
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species Cluster
1          5.1         3.5          1.4         0.2  setosa       1
2          4.9         3.0          1.4         0.2  setosa       1
3          4.7         3.2          1.3         0.2  setosa       1
4          4.6         3.1          1.5         0.2  setosa       1
5          5.0         3.6          1.4         0.2  setosa       1
6          5.4         3.9          1.7         0.4  setosa       1
>
> # Plot the clusters
> library(ggplot2)
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster)) +
+   geom_point(size = 3) +
+   labs(title = "K-Means Clustering of Iris Dataset", x = "Sepal Length", y = "
>
```

R Console
R Graphics: Device 2 (ACTIVE)
K-Means Clustering of Iris Dataset
Sepal Width
4.5
4.0
3.5
3.0
Cluster
1
2
3

RESULT:

Thus to implement clustering techniques (hierarchical and K Means) using R language is successfully completed.

Ex.No:10

VISUALIZE DATA USING ANY PLOTTING FRAMEWORK

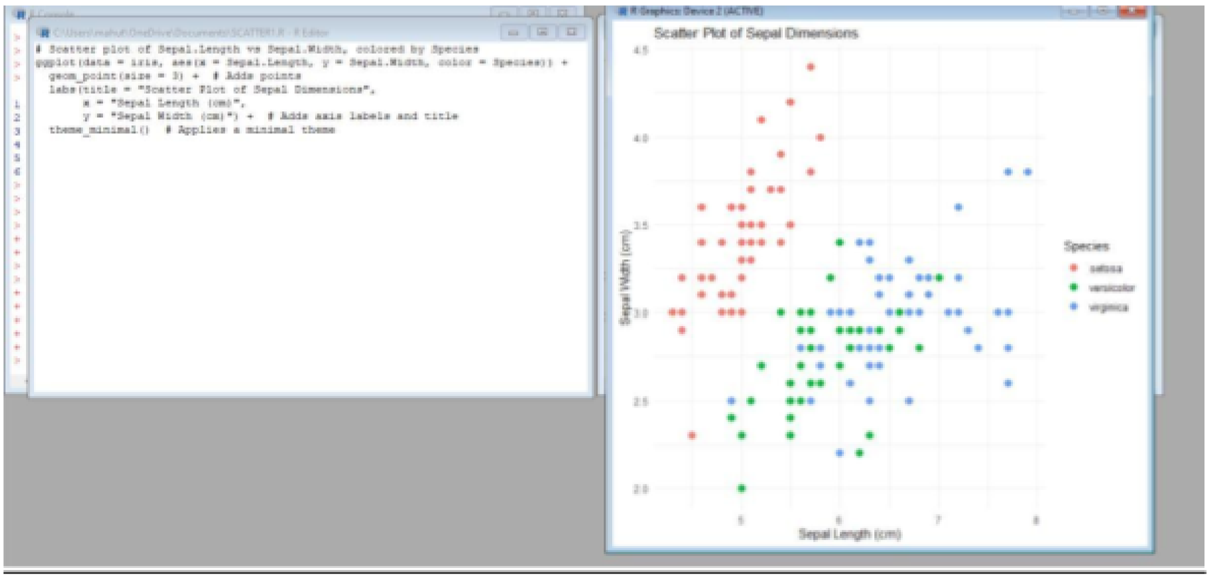
AIM:

To visualize data using any plotting framework using R language .

1) SCATTER PLOT

```
 # Install ggplot2 (if not already installed)
 install.packages("ggplot2")

 # Load the ggplot2 package
library(ggplot2)

 # Scatter plot of Sepal.Length vs Sepal.Width, colored by Species
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species))
+   geom_point(size = 3) +  # Adds points   labs(title = "Scatter Plot of Sepal
```

Dimensions", x = "Sepal Length (cm)", y = "Sepal Width (cm)") +



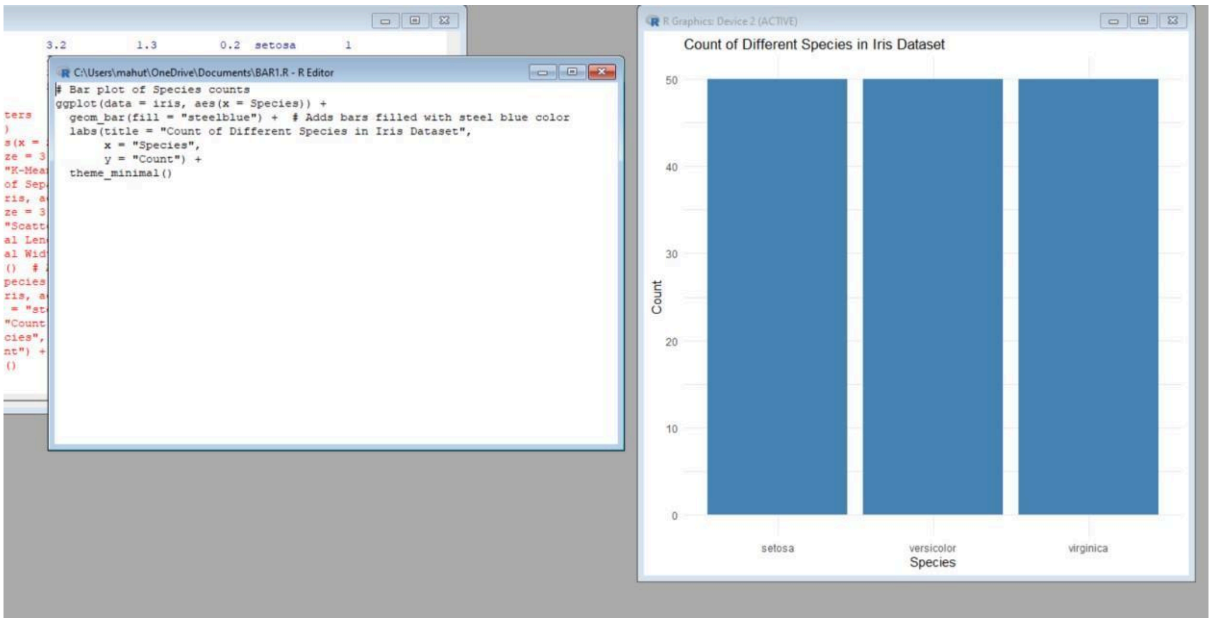
```
# Scatter plot of Sepal.Length vs Sepal.Width, colored by Species
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point(size = 3) +  # Adds points
  labs(title = "Scatter Plot of Sepal Dimensions",
       x = "Sepal Length (cm)",
       y = "Sepal Width (cm)") +  # Adds axis labels and title
  theme_minimal()  # Applies a minimal theme
```

C:\Users\mahuh\OneDrive\Documents\SCATTER1.R - R Editor
R Graphics: Device 2 (ACTIVE)
Scatter Plot of Sepal Dimensions
Sepal Width (cm)
Sepal Length (cm)
Species
setosa
versicolor
virginica

Adds axis labels and title theme_minimal() # Applies a minimal theme

2) <u>BAR CHART</u>

```
# Install ggplot2 (if not already installed)
install.packages("ggplot2")

# Load the ggplot2 package
library(ggplot2)

# Bar plot of Species counts
ggplot(data = iris, aes(x = Species)) +
  geom_bar(fill = "steelblue") +  # Adds bars filled with steel blue color
  labs(title = "Count of Different Species in Iris
Dataset",      x = "Species",      y = "Count") +
theme_minimal()
```



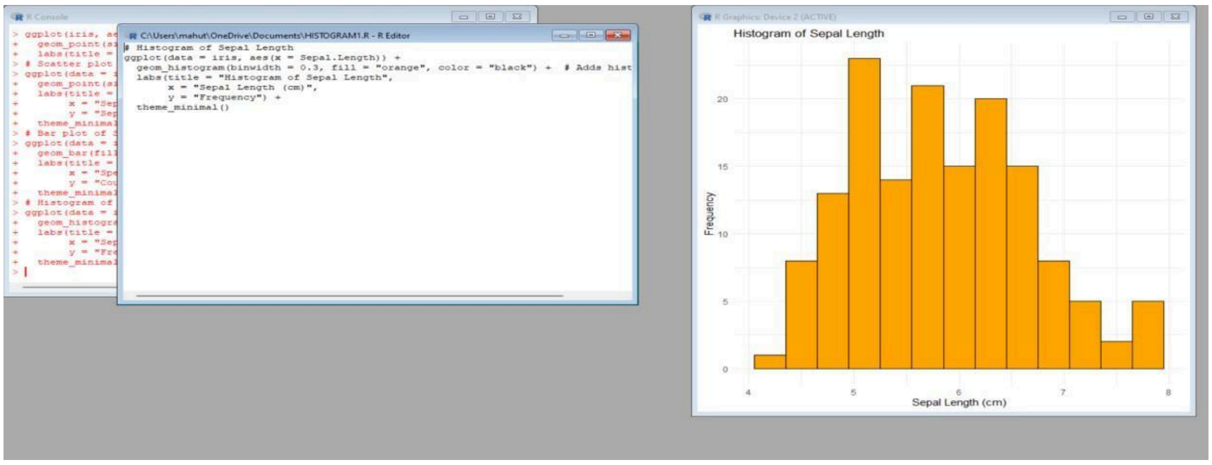
C:\Users\mahut\OneDrive\Documents\BAR1.R - R Editor

```
# Bar plot of Species counts
ggplot(data = iris, aes(x = Species)) +
  geom_bar(fill = "steelblue") +  # Adds bars filled with steel blue color
  labs(title = "Count of Different Species in Iris Dataset",
       x = "Species",
       y = "Count") +
  theme_minimal()
```

R Graphics: Device 2 (ACTIVE)
Count of Different Species in Iris Dataset
Count
Species
setosa
versicolor
virginica
0
10
20
30
40
50

3) HISTOGRAM

```
# Install ggplot2 (if not already installed)
install.packages("ggplot2")

# Load the ggplot2 package
library(ggplot2)

# Histogram of Sepal Length
ggplot(data = iris, aes(x = Sepal.Length)) +
  geom_histogram(binwidth = 0.3, fill = "orange", color = "black") +  # Adds
histogram bars
  labs(title = "Histogram of Sepal
Length",       x = "Sepal Length (cm)",
y = "Frequency") +
  theme_minimal()
```

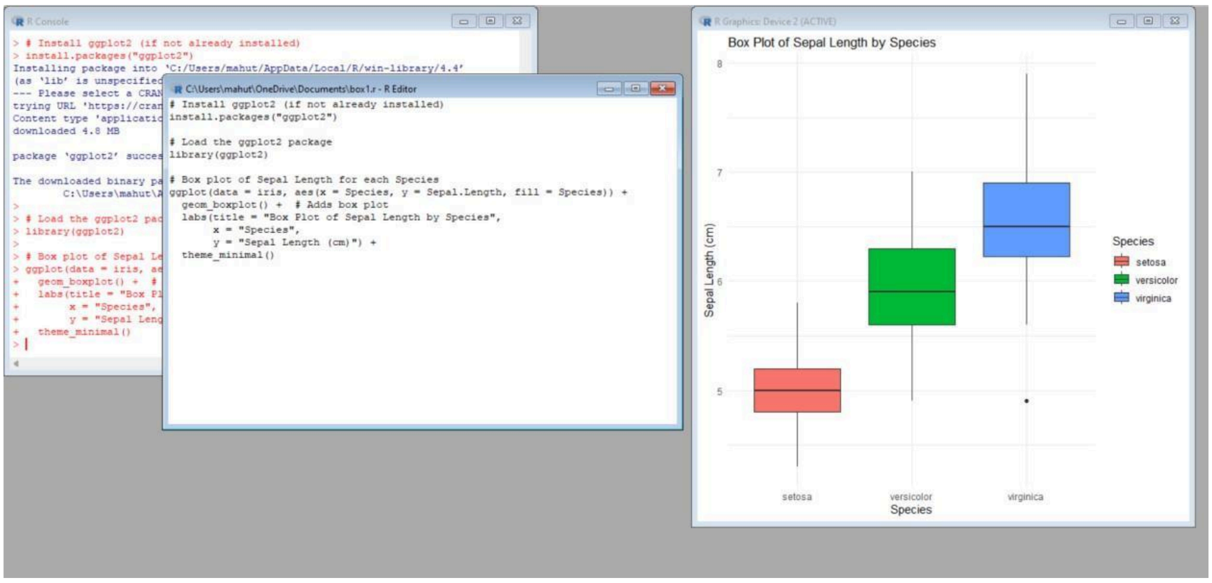


C:\Users\mahut\OneDrive\Documents\HISTOGRAM1.R - R Editor

```
# Histogram of Sepal Length
ggplot(data = iris, aes(x = Sepal.Length)) +
  geom_histogram(binwidth = 0.3, fill = "orange", color = "black") +  # Adds hist
  labs(title = "Histogram of Sepal Length",
       x = "Sepal Length (cm)",
       y = "Frequency") +
  theme_minimal()
```

4)BOX PLOT

```
# Install ggplot2 (if not already installed)
install.packages("ggplot2")

# Load the ggplot2 package
library(ggplot2)

# Box plot of Sepal Length for each Species
ggplot(data = iris, aes(x = Species, y = Sepal.Length, fill =
Species)) +   geom_boxplot() +  # Adds box plot   labs(title = "Box
Plot of Sepal Length by Species",        x = "Species",        y = "Sepal
Length (cm)") +   theme_minimal()
```


R Console
C:\Users\mahut\OneDrive\Documents\box1.r - R Editor
R Graphics: Device 2 (ACTIVE)

```
> # Install ggplot2 (if not already installed)
> install.packages("ggplot2")
Installing package into 'C:/Users/mahut/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified
--- Please select a CRAN
trying URL 'https://cran
Content type 'applicatio
downloaded 4.8 MB

package 'ggplot2' succes

The downloaded binary pa
        C:\Users\mahut\A
>
> # Load the ggplot2 pac
> library(ggplot2)
>
> # Box plot of Sepal Le
> ggplot(data = iris, ae
+   geom_boxplot() +  #
+   labs(title = "Box Pl
+        x = "Species",
+        y = "Sepal Leng
+   theme_minimal()
> 
```

```
# Install ggplot2 (if not already installed)
install.packages("ggplot2")

# Load the ggplot2 package
library(ggplot2)

# Box plot of Sepal Length for each Species
ggplot(data = iris, aes(x = Species, y = Sepal.Length, fill = Species)) +
  geom_boxplot() +  # Adds box plot
  labs(title = "Box Plot of Sepal Length by Species",
       x = "Species",
       y = "Sepal Length (cm)") +
  theme_minimal()
```

RESULT :
Thus to visualize data using plotting framework using R language is successfully completed.