

WELCOME

STATISTICAL DATA ANALYSIS ON IRIS DATASET USING R



Project Guide:
K NAVEEN KUMAR

Project by:
G SHIVA RAM

AGENDA



- ☐ Introduction to Iris Data
- ☐ About data set
- ☐ Data collection
- ☐ Overview of the dataset
- ☐ About software
- ☐ Methodology
- ☐ Data visualization
- ☐ conclusion

Introduction to Iris data



Sample Data

Observations	Variables				
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
	5.1	3.5	1.4	0.2	setosa
	4.9	3	1.4	0.2	setosa
	4.7	3.2	1.3	0.2	Versicolor
	4.6	3.1	1.5	0.2	Virginica
	5	3.6	1.4	0.2	Virginica

Why Iris Data?

The Iris Data set was introduced by Robert Fisher. It consists of four measures i.e. Length and Width of Sepals and Petals for three flower species (namely Setosa, Versicolor and Virginica).

1. Easily available and lot of support material available online
2. All types of machine learning algorithms can be easily implemented on this data
3. Caution : Be aware that the results on real-life datasets are not going to be as optimistic.

ABOUT THE IRIS DATA



The iris dataset contains the following data

We have 50 samples of 3 different species of iris (150 samples total)

- Here we can see that given 4 features i.e sepal length, sepal width, petal length, and petal width determine whether a flower is Setosa, Versicolor or Virginica.
- Sepal length, Sepal width, Petal length, Petal width are called feature/Independent-variable.
- Species are called Labels/Dependent-variable.

TYPES OF IRIS FLOWERS



iris setosa



petal

sepal

iris versicolor



petal

sepal

iris virginica



petal

sepal

ATTRIBUTE INFORMATION



- Four features of flower: **length** and the **width** of sepal and petal

Sepal length in cm

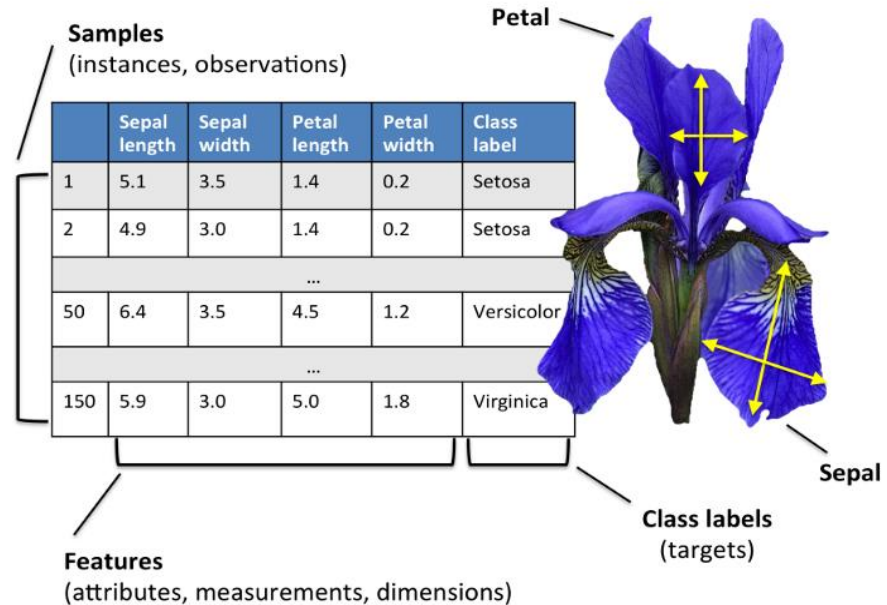
Sepal width in cm

Petal length in cm

Petal width in cm

class:

- 1) Iris Setosa
- 2) Iris Versicolour
- 3) Iris Virginica



Data Collection



- **UC Irvine Machine Learning Repository**

If you want to download the data set instead of using the one that is built into R, you can go to the [UC Irvine Machine Learning Repository](#) and look up the Iris data set.

Overview Of The Data Set



- First, you can already try to get an idea of your data by making some graphs, such as boxplots. In this case, however, scatter plots can give you a great idea of what you're dealing with: it can be interesting to see how much one variable is affected by another.
- In other words, you want to see if there is any correlation between two variables.
- Review basic descriptive statistics in R
- To display summary statistics for each feature available in dataset.
- The linear combination of original variables that provide the best possible separation between the groups.

About Software



What is R & Why we use it.

- It's a tool : Open-Source, cross platform, free programming language designed to build statistical solutions
- Powerful : Gives access to CRAN repository containing over 10,000 packages with pre-defined functions for almost every purpose
- Stays Relevant : Constantly being updated by users (Scientists, Statisticians, Researchers, Students!)
- More: Makes beautiful graphs, can create custom functions or modify existing ones, can be integrated into many environments and platforms such as Hadoop etc.

Installing R



- Download the version compatible with your OS
- Can be downloaded for free from
<http://www.r-project.org/>
- Simple/Standard installation process

Installing R -Studio :

- Can be downloaded for free from:
<https://www.rstudio.com/products/rstudio/download/>
- Download the free version compatible with your OS
- R needs to be installed before installing R- Studio

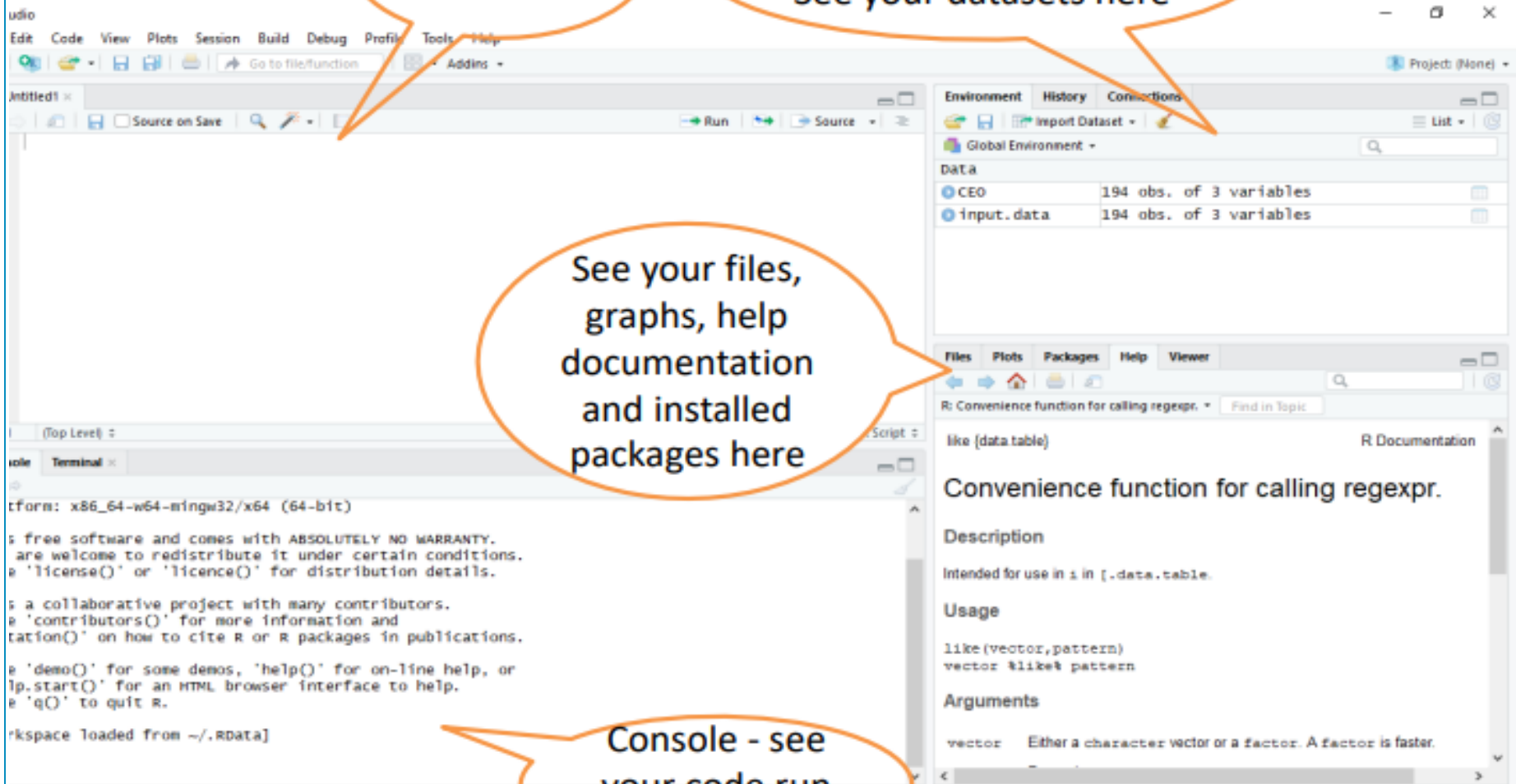
R-Studio UI

Write your
code here

Global Environment-
See your datasets here

See your files,
graphs, help
documentation
and installed
packages here

Console - see
your code run
here



Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use
or distribution prohibited

METHODOLOGY



- Enter the data in excel sheet & save the data with the **read.csv()**– It is used to read csv files and create a data frame from it.
- We import iris data by giving path of data file of “iris.csv”.
- Iris = **read.csv**(“iris.csv”)
- Import the data from excel to R & Explore the *iris* Dataset with R.
- Let's get started by importing all the libraries that we are going to need.
- Importing Packages: **"ggplot2", "dplyr", "MASS"**.

DATA EXPLORATION AND VISUALISATION



Some basic function in R to examine iris dataset:

➤ `data(iris)`

	A	B	C	D	E	F
1	sepal_length	sepal_width	petal_length	petal_width	species	
2	5.1	3.5	1.4	0.2	setosa	
3	4.9	3	1.4	0.2	setosa	
4	4.7	3.2	1.3	0.2	setosa	
5	4.6	3.1	1.5	0.2	setosa	
6	5	3.6	1.4	0.2	setosa	
7	5.4	3.9	1.7	0.4	setosa	
8	4.6	3.4	1.4	0.3	setosa	
9	5	3.4	1.5	0.2	setosa	
10	4.4	2.9	1.4	0.2	setosa	
11	4.9	3.1	1.5	0.1	setosa	
12	5.4	3.7	1.5	0.2	setosa	
13	4.8	3.4	1.6	0.2	setosa	
14	4.8	3	1.4	0.1	setosa	
15	4.3	3	1.1	0.1	setosa	
16	5.8	4	1.2	0.2	setosa	
17	5.7	4.4	1.5	0.4	setosa	
18	5.4	3.9	1.3	0.4	setosa	
19	5.1	3.5	1.4	0.3	setosa	
20	5.7	3.8	1.7	0.3	setosa	
21	5.1	3.8	1.5	0.3	setosa	



	A	B	C	D	E
1	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
2	5.1	3.5	1.4	0.2	Iris-setosa
3	4.9	3	1.4	0.2	Iris-setosa
4	4.7	3.2	1.3	0.2	Iris-setosa
5	4.6	3.1	1.5	0.2	Iris-setosa
6	5	3.6	1.4	0.2	Iris-setosa
7	5.4	3.9	1.7	0.4	Iris-setosa
8	4.6	3.4	1.4	0.3	Iris-setosa
9	5	3.4	1.5	0.2	Iris-setosa
10	4.4	2.9	1.4	0.2	Iris-setosa
11	4.9	3.1	1.5	0.1	Iris-setosa
12	5.4	3.7	1.5	0.2	Iris-setosa
13	4.8	3.4	1.6	0.2	Iris-setosa
14	4.8	3	1.4	0.1	Iris-setosa
15	4.3	3	1.1	0.1	Iris-setosa
16	5.8	4	1.2	0.2	Iris-setosa
17	5.7	4.4	1.5	0.4	Iris-setosa
18	5.4	3.9	1.3	0.4	Iris-setosa
19	5.1	3.5	1.4	0.3	Iris-setosa
20	5.7	3.8	1.7	0.3	Iris-setosa
21	5.1	3.8	1.5	0.3	Iris-setosa
22	5.4	3.4	1.7	0.2	Iris-setosa
23	5.1	3.7	1.5	0.4	Iris-setosa
24	4.6	3.6	1	0.2	Iris-setosa
25	5.1	3.3	1.7	0.5	Iris-setosa

attributes() :It shows attributes of iris data

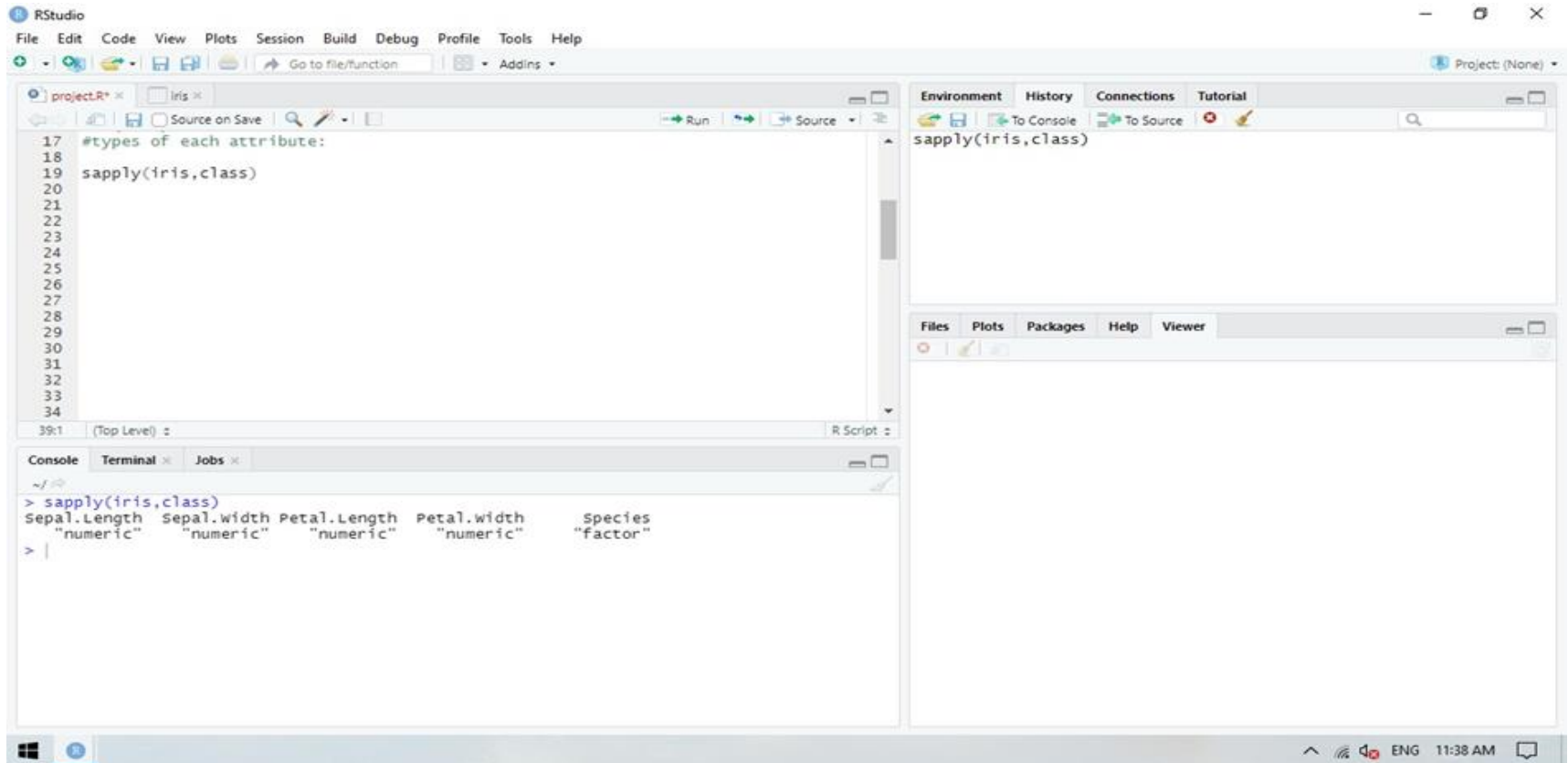
```
> attributes(iris)
$names
[1] "Sepal.Length" "Sepal.width"  "Petal.Length" "Petal.width"  "Species"

$class
[1] "data.frame"

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


sapply():

sapply() function takes list, vector or data frame as input and gives output in vector or matrix. It is useful for operations on list objects and returns a list object of same length of original set : **sapply (iris, class)**



The screenshot shows the RStudio interface with the following components:

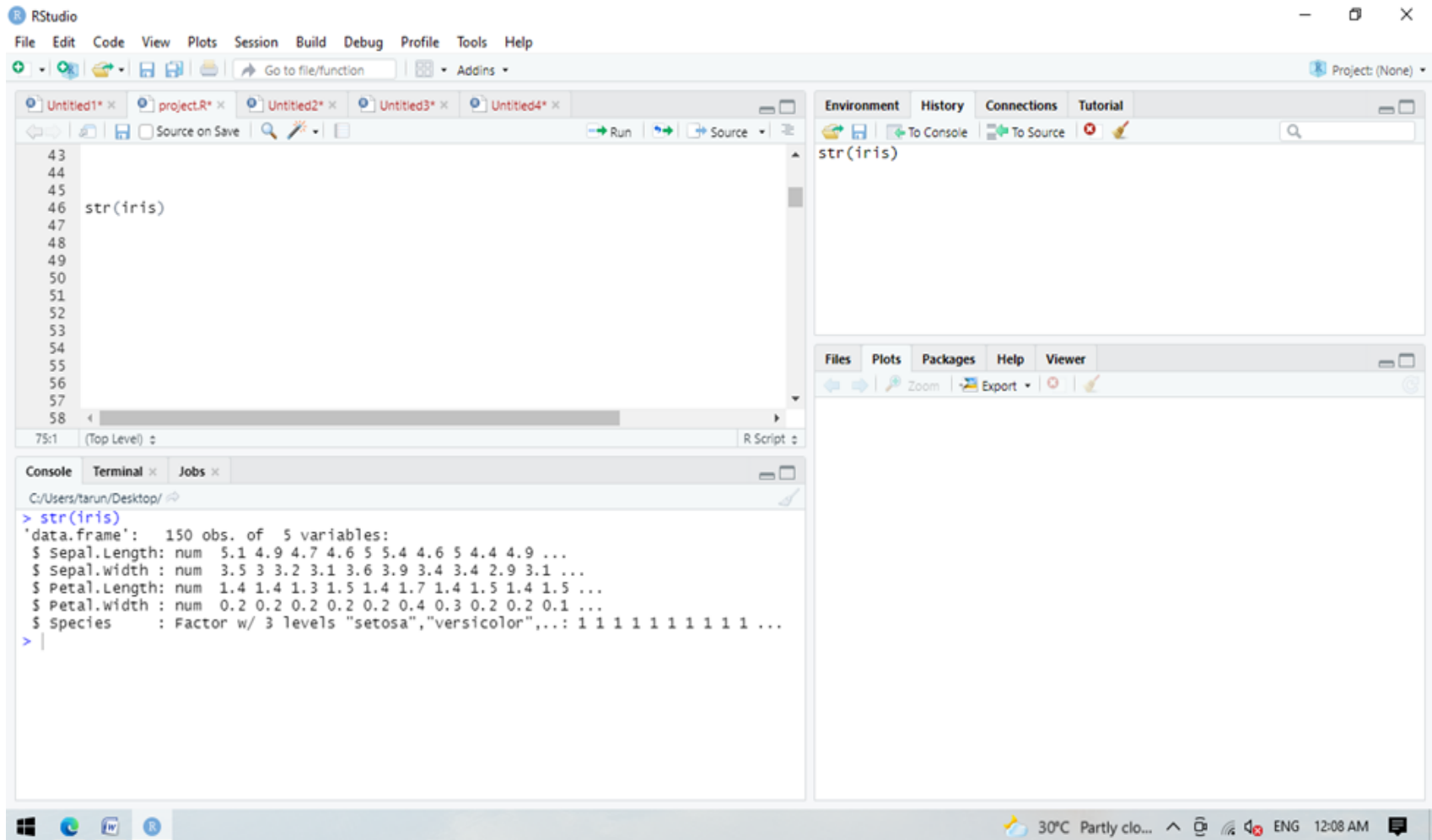
- Source Editor:** Contains the R script:

```
17 #types of each attribute:
18
19 sapply(iris,class)
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
```
- Environment Panel:** Shows the expression `sapply(iris,class)` that has been executed.
- Console:** Displays the output of the `sapply` function:

```
> sapply(iris,class)
Sepal.Length Sepal.width Petal.Length Petal.width  Species
"numeric"    "numeric"    "numeric"    "numeric"    "factor"
```

The status bar at the bottom indicates the system is running Windows, the language is set to English, and the time is 11:38 AM.

str(iris): It is used to give the structure of the data



The screenshot shows the RStudio interface. The source editor on the left contains the R code `str(iris)` at line 46. The console at the bottom displays the output of this command, showing the structure of the `iris` data frame. The Environment pane on the right shows `str(iris)` as the current object. The status bar at the bottom indicates the system temperature is 30°C and the time is 12:08 AM.

```
43  
44  
45  
46 str(iris)  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58
```

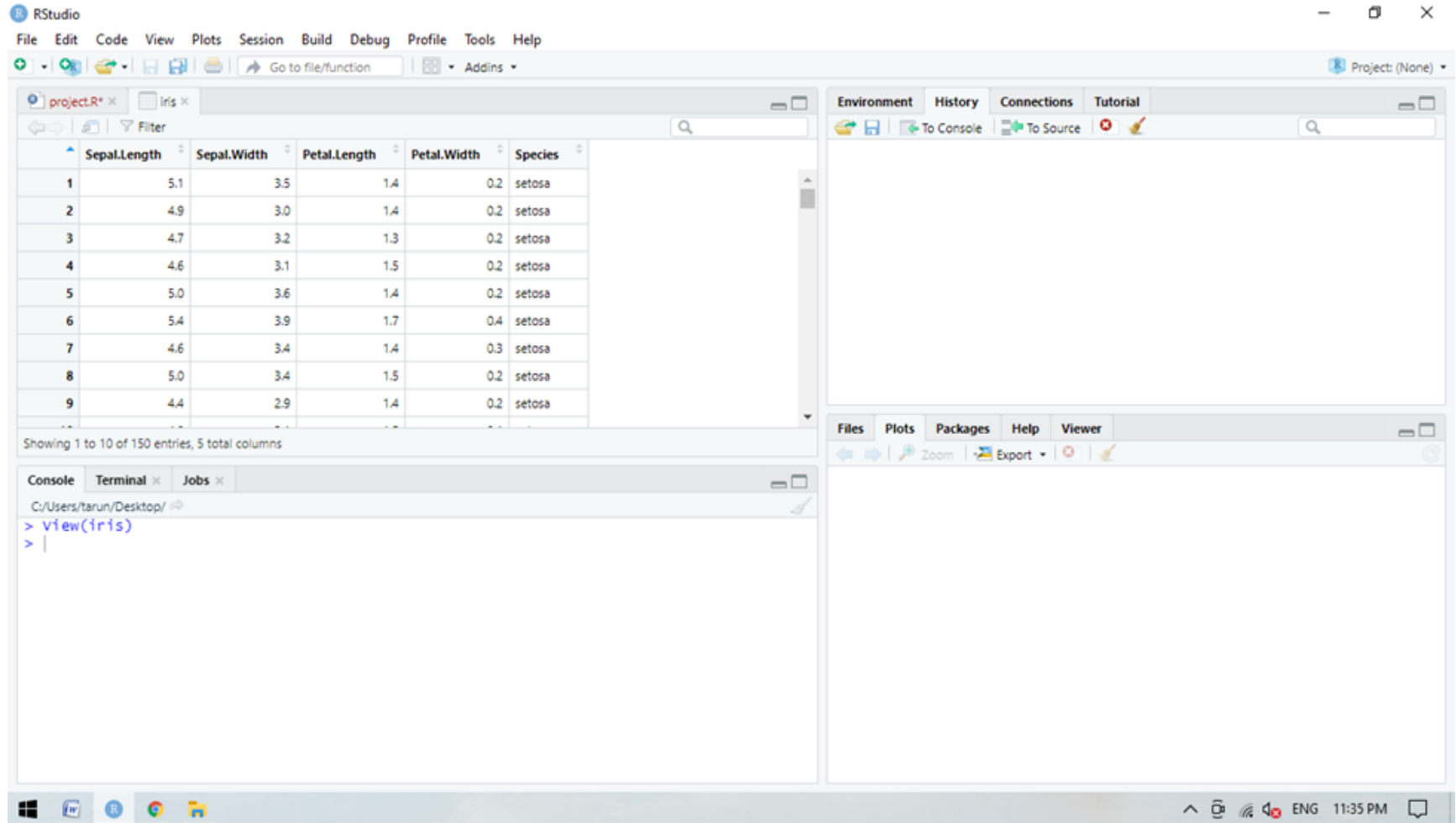
Environment History Connections Tutorial
To Console To Source

Files Plots Packages Help Viewer
Zoom Export

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

30°C Partly clo... ENG 12:08 AM

View(): To View the data : View(iris)



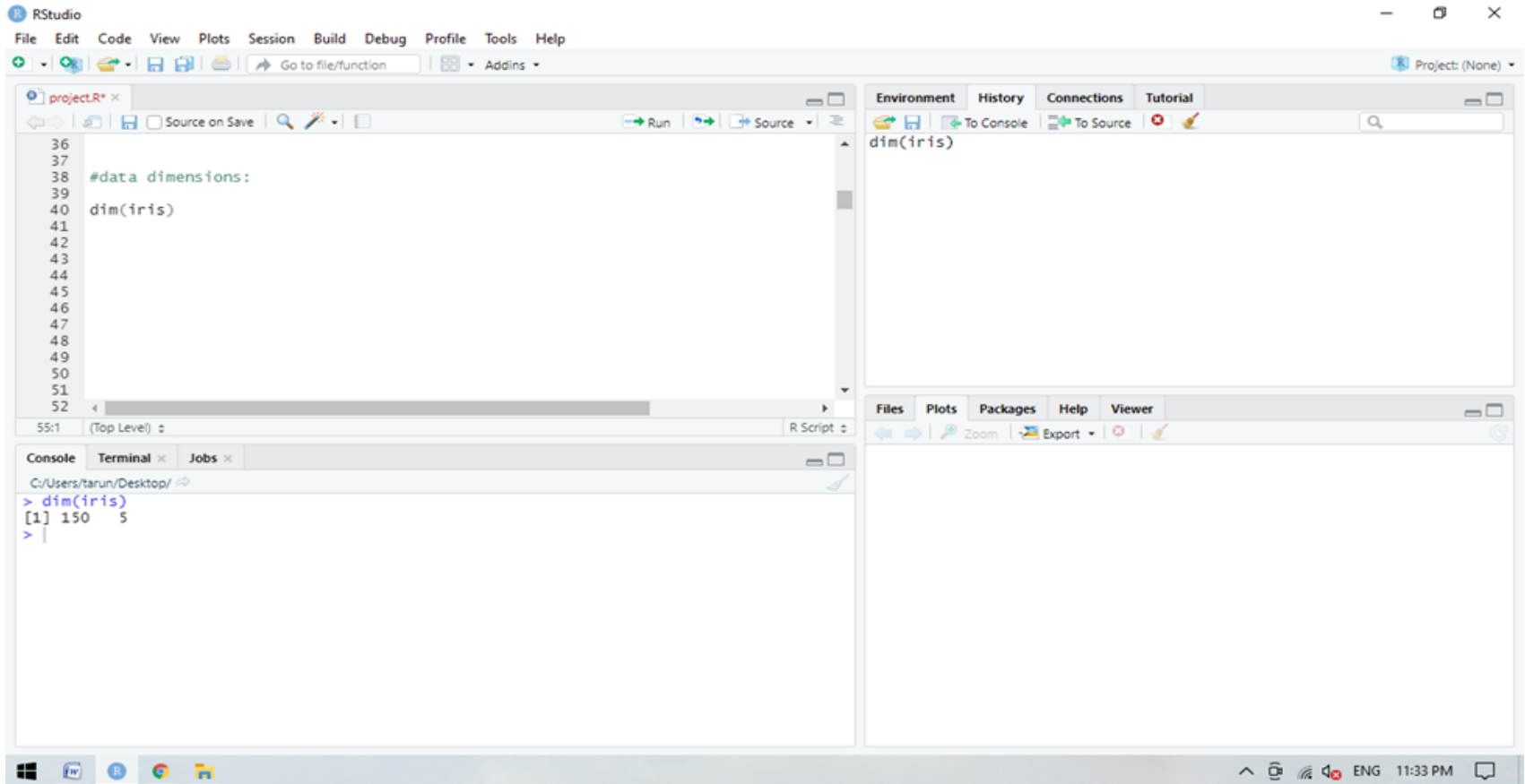
The screenshot shows the RStudio interface with the following components:

- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for saving, opening, and navigating files, along with a search bar.
- Environment Panel:** Shows the current environment with variables like 'iris'.
- Console:** Displays the command `> view(iris)` and the current directory `C:/Users/tarun/Desktop/`.
- Data Viewer:** Displays a table of the first 10 rows of the 'iris' dataset.

	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
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa

Showing 1 to 10 of 150 entries, 5 total columns

dim(): We can get a quick idea of how many instances (rows) and how many attributes (columns) the data contains with the **dim** function : `dim(iris)`



The screenshot shows the RStudio interface. The source editor on the left contains the following code:

```
36  
37  
38 #data dimensions:  
39  
40 dim(iris)  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52
```

The console at the bottom shows the execution of the command:

```
> dim(iris)  
[1] 150 5  
>
```

The Environment pane on the right shows the variable `dim(iris)` as a list.

head(iris); tail(iris)

The image shows the RStudio interface with the following components:

- Source Editor:** Contains R code for viewing the first and last rows of the iris dataset.

```
22 #To view the first & last 6 rows of the data:
23
24 head(iris)
25 tail(iris)
26
27
28
29
30
31
32
33
34
35
```
- Environment:** Shows the objects created in the environment: `head(iris)` and `tail(iris)`.
- Console:** Displays the output of the commands.

```
> head(iris)
  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

> tail(iris)
  Sepal.Length Sepal.width Petal.Length Petal.width Species
145          6.7         3.3         5.7         2.5 virginica
146          6.7         3.0         5.2         2.3 virginica
147          6.3         2.5         5.0         1.9 virginica
148          6.5         3.0         5.2         2.0 virginica
149          6.2         3.4         5.4         2.3 virginica
150          5.9         3.0         5.1         1.8 virginica
```

The status bar at the bottom indicates the system is running Windows, the user is 'tarun', and the time is 09:41 PM.

summary()

The `summary()` function in R is used to obtain the summary statistics of the dataset, including minimum value, 1st quantile, median, mean 3rd quantile, maximum value for each numerical variable and the count for each level of the only categorical variable “species”.

which is displayed in output : **summary(iris)**

```
Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
Min.   :4.300  Min.   :2.000  Min.   :1.000  Min.   :0.100
1st Qu.:5.100  1st Qu.:2.800  1st Qu.:1.600  1st Qu.:0.300
Median :5.800  Median :3.000  Median :4.350  Median :1.300
Mean   :5.843  Mean   :3.057  Mean   :3.758  Mean   :1.199
3rd Qu.:6.400  3rd Qu.:3.300  3rd Qu.:5.100  3rd Qu.:1.800
Max.   :7.900  Max.   :4.400  Max.   :6.900  Max.   :2.500
Species
setosa   :50
versicolor:50
virginica :50
```

VISUALIZE DATASET, SPLIT THE DATA

```
x=iris[,1:4]
```

```
View(x)
```

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains the R code `x=iris[,1:4]` and `View(x)`.
- Environment Panel:** Shows the variable `x` as a data frame with 12 rows and 4 columns.
- Viewer Panel:** Displays a preview of the data frame `x`.
- Console:** Shows the execution of the R code, with the prompt `>` and the output `x=iris[,1:4]` and `View(x)`.
- Table:** A table with 12 rows and 4 columns, showing the first 12 entries of the dataset. The columns are labeled `Sepal.Length`, `Sepal.Width`, `Petal.Length`, and `Petal.Width`.

	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

`y=iris[,5]`
View(y)

The screenshot shows the RStudio interface with the following components:

- Top Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Top Toolbar:** Includes icons for saving, opening, and navigating files, along with a search bar and an "Addins" dropdown.
- Left Panel (Environment):** Displays a list of objects. The object "y" is selected, showing its contents as a vector of 12 "setosa" values. Below the list, it says "Showing 1 to 13 of 150 entries, 1 total columns".
- Right Panel (Environment/History):** Shows the R console output for the commands:

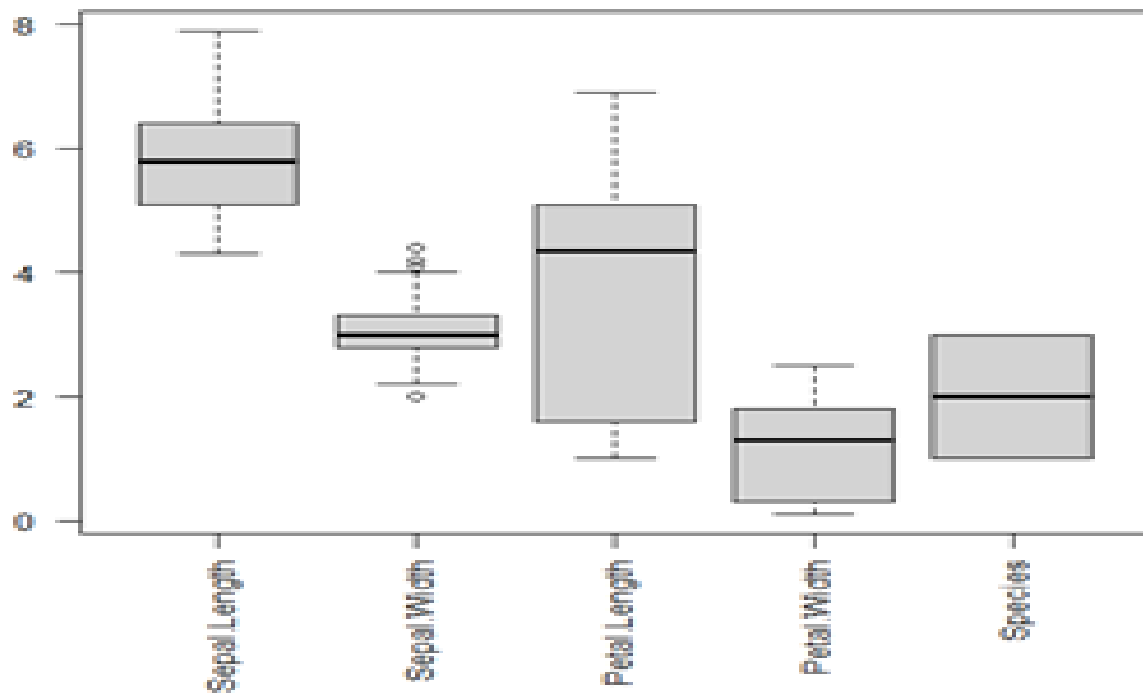
```
x=iris[,1:4]
view(x)
y=iris[,5]
view(y)
```
- Bottom Panel (Console):** Shows the R prompt with the commands:

```
> y=iris[,5]
> view(y)
> |
```
- Bottom Status Bar:** Displays system information including the Windows taskbar, network status, and the time "12:29 AM".

BOX PLOTS



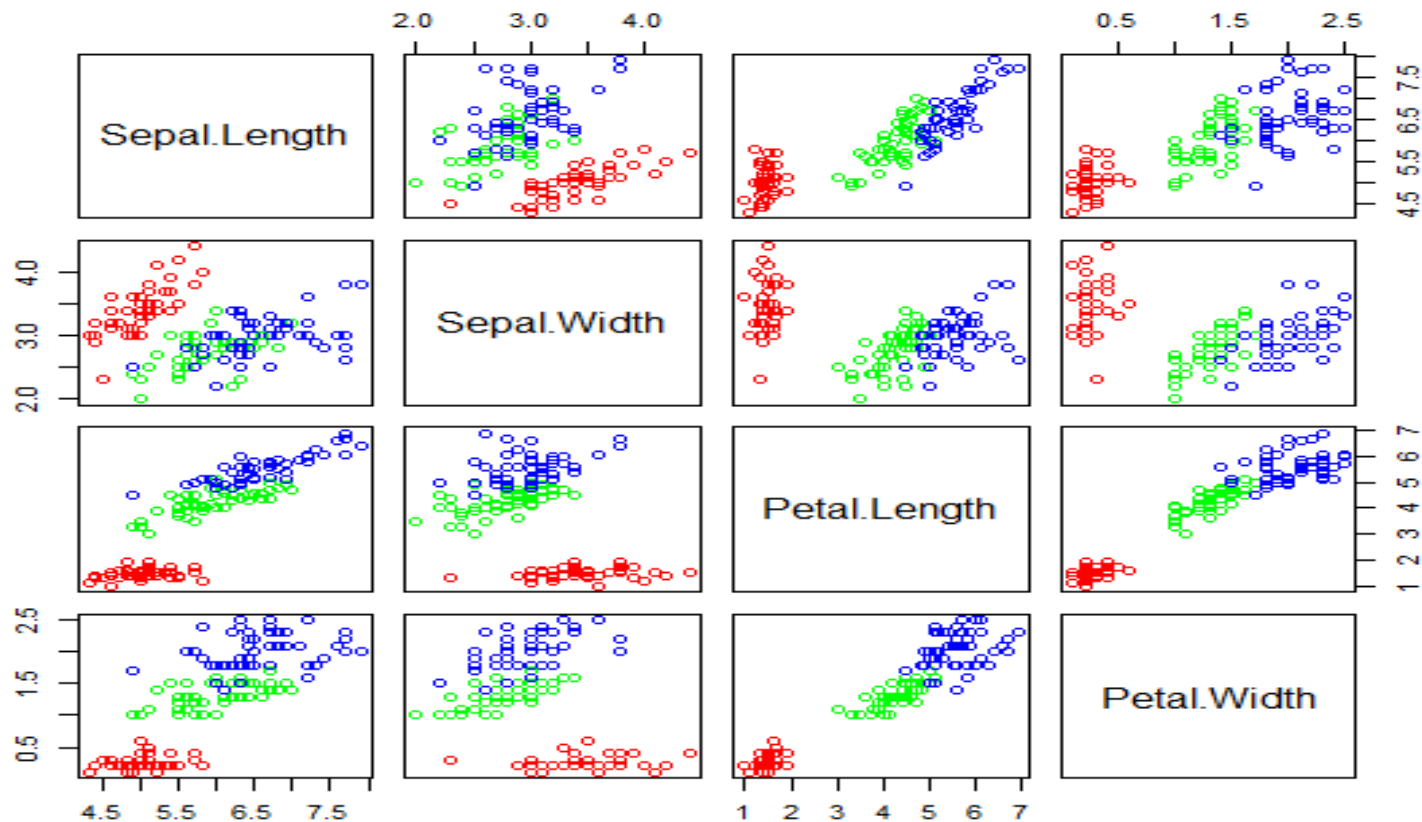
- Box plot, also known as a box and whisker plot, displays a summary of a large amount of data in five numbers - minimum, lower quartile(25th percentile), median(50th percentile), upper quartile(75th percentile) and maximum data values.



SCATTER PLOT



Scatter plots are used to plot data points on a horizontal and a vertical axis in the attempt to show how much one variable is affected by another. The relationship between two variables is called their correlation.



From the below plot, there seems to be a positive correlation between the length and width of all the species, however there is a distinguishing strong correlation and relationship between petal length and petal width.

LINEAR DISCRIMINANT ANALYSIS



Linear Discriminant Analysis(LDA) Linear discriminant methods group images of the same classes and separates images of the different classes. To identify an input test image, the projected test image is compared to each projected training image, and the test image is identified as the closest training image.

- ❖ Let's create a training dataset and test dataset for prediction and testing purposes. 60% dataset used for training purposes and 40% used for testing purposes.

```
> linear=lda(Species~.,training);linear
```

```
Call:
```

```
lda(Species ~ ., data = training)
```

```
Prior probabilities of groups:
```

	setosa	versicolor	virginica
	0.3837209	0.3139535	0.3023256

```
Group means:
```

	Sepal.Length	Sepal.width	Petal.Length	Petal.width
setosa	4.975758	3.357576	1.472727	0.2454545
versicolor	5.974074	2.751852	4.281481	1.3407407
virginica	6.580769	2.946154	5.553846	1.9807692

```
Coefficients of linear discriminants:
```

	LD1	LD2
Sepal.Length	1.252207	-0.1229923
Sepal.width	1.115823	2.2711963
Petal.Length	-2.616277	-0.7924520
Petal.width	-2.156489	2.6956343

```
Proportion of trace:
```

	LD1	LD2
	0.9937	0.0063

- Based on the training dataset, 38% belongs to setosa group, 31% belongs to versicolor groups and 30% belongs to virginica groups.
- The proportion of trace tells us how well each discriminant distinguishes between the species, and given the very high size of the first discriminant (0.9937), we see that LD1 explains 99% of the variance and so LD2 is contributing very little to the distinction of species.

CONCLUSION



- While Setosa can be easily identified (Linearly separable), Virginica and Versicolor have some overlap (almost linearly separable).
- The percentage separation achieved for the first discriminant function is 99.3%. so we have clear separation among the three species.
- The separation percentage achieved by LD2 is not good. That is quietly less than one percent.



THANK YOU