|  |
| --- |
| UMUC |
| Using R to Explore Iris Data |
| Module 1 Exercise 1 |
|  |
| **DATA 630- Data Mining** |
|  |

|  |
| --- |
| In this exercise, you will get familiar with R-Studio interface, and will use an interface to explore data in the iris dataset that comes with software installation. The data exploration includes previewing the data, descriptive statistics, grouping, frequency tables, and data visualization. Before working on this exercise, you need to install the latest version of R and R Studio on your PC. |

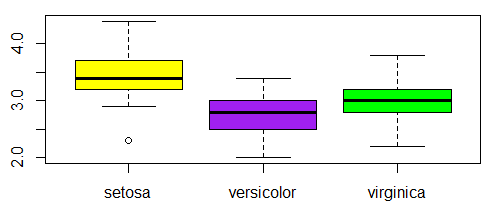


Table of Contents

[1. Get Started 2](#_Toc512283840)

[2. Launch the R Studio Program 2](#_Toc512283841)

[3. Get familiar with R Studio Interface 2](#_Toc512283842)

[4. Open the Iris Data R Script in Source Editor 4](#_Toc512283843)

[5. Checking the R version and System Information 4](#_Toc512283844)

[6. Getting help 6](#_Toc512283845)

[7. Exploring Iris Dataset 7](#_Toc512283846)

[Iris Data Help Page 7](#_Toc512283847)

[Previewing Iris Data and Exploring its Structure 8](#_Toc512283848)

[Descriptive Statistics 13](#_Toc512283849)

[Data Visualization 14](#_Toc512283850)

[Data Aggregation 17](#_Toc512283851)

[8. How to Create New Script 19](#_Toc512283852)

[9. Exit R Studio 20](#_Toc512283853)

R Studio Introduction and Iris Data Exploration

# Get Started

In this tutorial, you will get familiar with RStudio interface and will explore the pre-installed iris data. Before working on this tutorial, you need to

1. Install the latest R and RStudio version. The installation requires the PC administrative rights. You need to login to your PC as an administrator or enter the administrator credentials when prompted during the installation.
2. Download the Exploring Iris Data.R file from Exercise 1 assignment folder.

# Launch the R Studio Program

Select RStudio from the start menu on to launch the program. You may also double-click on RStudio Shortcut if you created one.

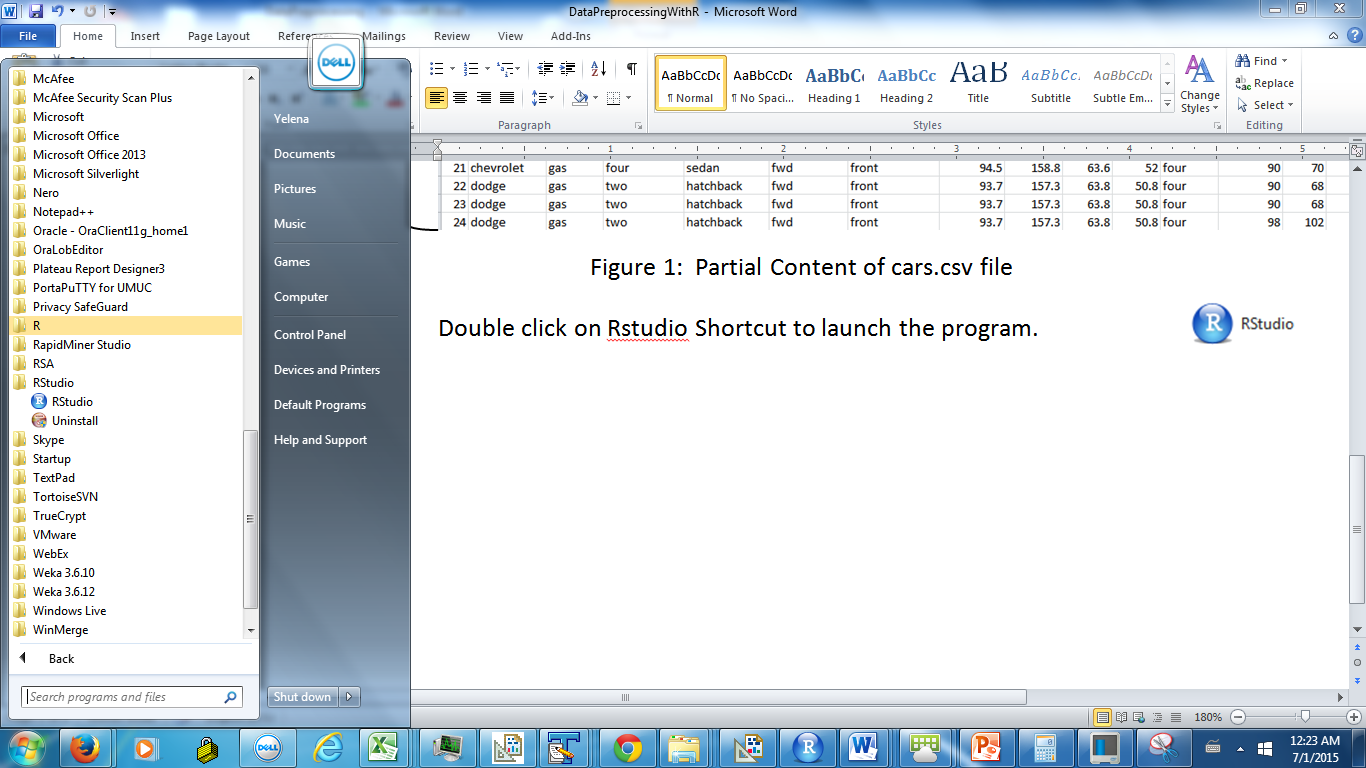
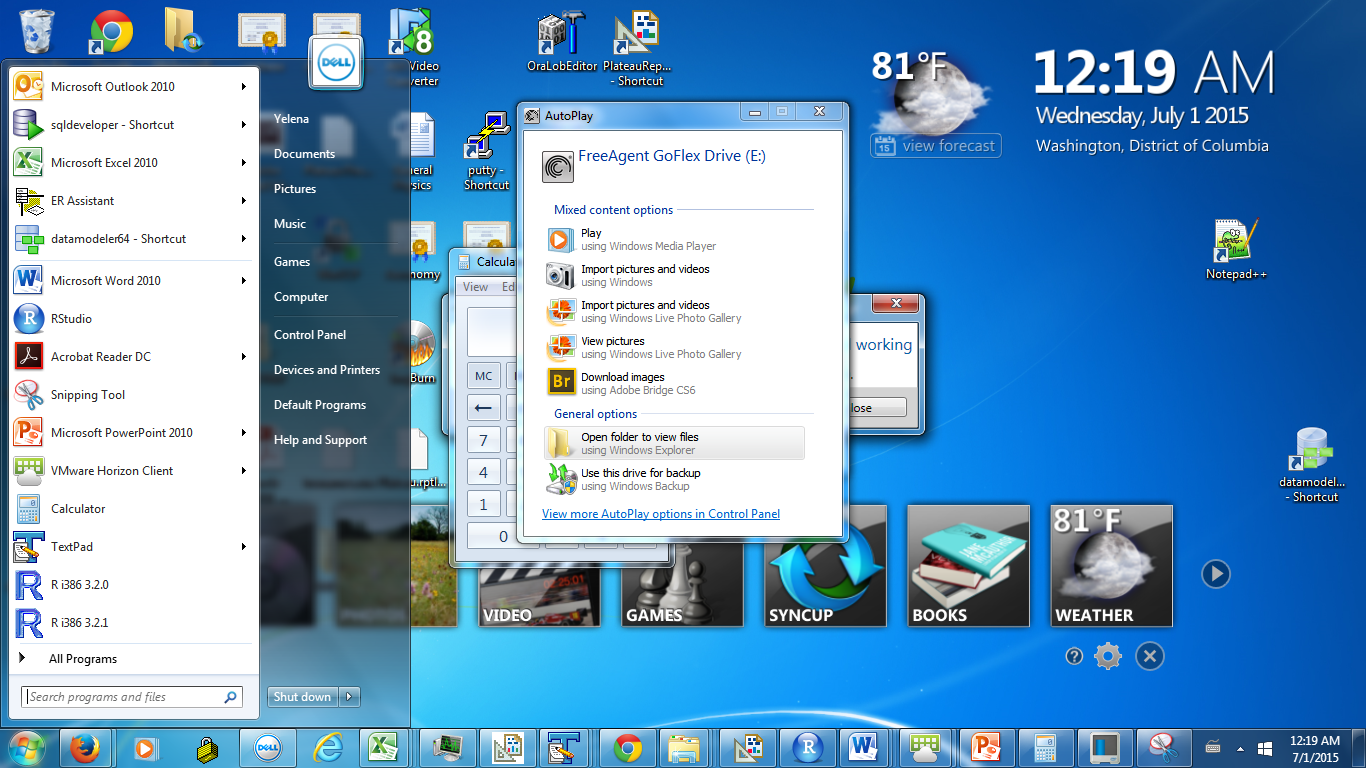


Figure 1: Select RStudio from the menu

# Get familiar with R Studio Interface

An R Studio interface in Figure 2 will open. By default, the interface contains 3 main resizable panels. The left panel is a **console window**.

* When you launch the program, the text in the console window includes an R version.
* You may type the commands at the > prompt.
* The console window also displays the command output and the error messages.
* Use up and down arrow keys to navigate the commands history.

Keep in mind that R language is **case sensitive**. For instance, Dir and dir commands have a different meaning.

The number of spaces inside the commands does not matter. However, some spacing may improve readability.

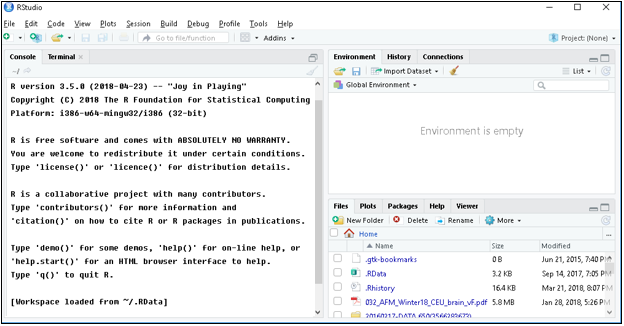
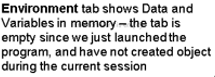
The **top right panel** contains the environment tab and history tab.

* The **environment tab** shows the variables, functions, and data in memory created during the current session. The current session starts when you launch the program. Initially, the environment tab is empty.
* The **History tab** allows viewing the commands you ran during the current session. Each time you run a command, a new entry is added to the history. However, the commands are not saved to the file automatically, and all unsaved history will be lost at the end of the session.

Figure 3 shows the content of the history tab after the commands have been executed in the current session. To rerun the command from the history, highlight the command and click To Console.

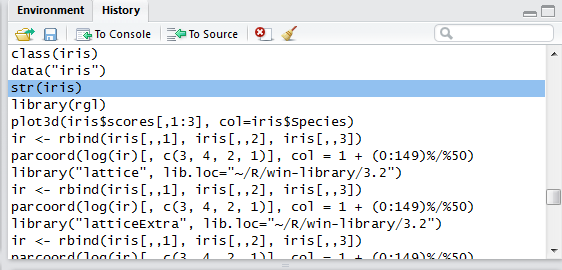
The **bottom right** panel contains the files tab, plots tab, packages tab, and help tab.

* The **files tab** shows the files in the current working directory. A **working directory** is a hard drive location where R looks for a file that you attempt to open.
* The **plots tab** displays the charts you build and supports navigation between charts.
* The **packages tab** lists the R packages that contain the functions/commands implementation. Some packages that you will use in this course will need to be installed separately from R Studio installation, and the instructions will be provided.
* The **help tab** provides the information on commands, including the input parameters, the output parameters, and examples.



The R version

Figure 2: R Studio Interface



Use the broom icon to clear the entire history

To rerun the command from the history:

1. Highlight the command you want to rerun.
2. Click To Console

Figure 3: History Tab

# Open the Iris Data R Script in Source Editor

Although you may enter the commands in the console window, you will be using a source editor to write and execute the R scripts. Using the source editor instead of console window has the following advantages:

* The work is reproducible – to rerun the sequence of commands, you rerun the script
* Code reuse – you may reuse portions of the code for the next assignment
* Code documentation – you may add descriptive comments to the code
* Code sharing - some exercise code is shared with you
* May execute multiple lines at a time

To open the Exploring Iris Data.R Script, select Open File from the File menu and find the file downloaded from Exercise 1 assignment folder.

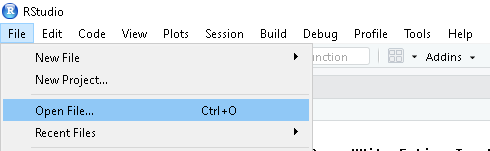


Figure 4: Select Open File

The script will open in a source window at the top left in Figure 5.

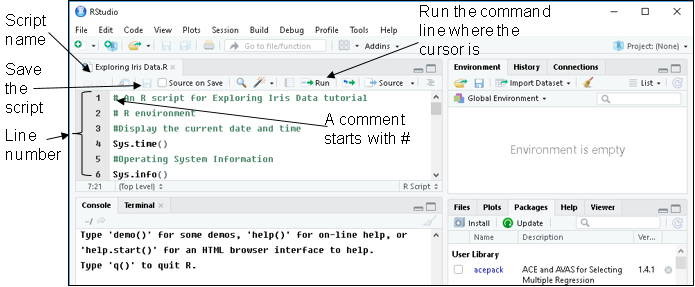


Figure 5: The Exploring Iris data code in a source window

# Checking the R version and System Information

Figure 6 shows lines 1-8 in the Exploring Iris Data script.

The first line in the file starts from # character. It means that the rest of the line is a code **comment**. A comment is a non-executable line of code. The comments are useful for documenting the code.

The first executable command on line 4 displays the current time. Place a cursor on line 4 and click on the Run button on the source window toolbar.

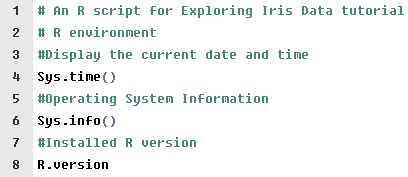


Figure 6: Lines 1-8 from Exploring iris data script

The command and the comments above it will be echoed in the console window. Figure 7 shows the command we ran and the command output in the console window.

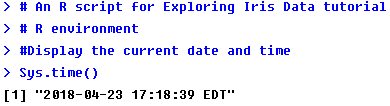


Figure 7: Sys.time() command output

The Sys.info() command on line 6 returns the R Studio version and operating system information. Place the cursor on line 6 and click on Run command at the top toolbar.

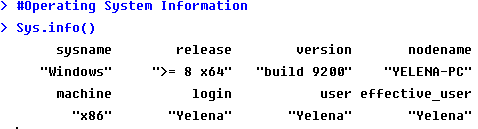


Figure 8: System Information

The R.version command on line 8 displays the information on the installed R version. Run the command and verify that you are using the latest R version.

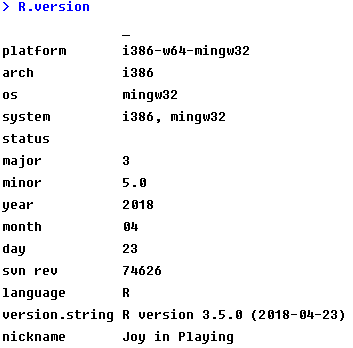


Figure 9: The Installed R Version

# Getting help

We use the help command to display a help page for a command in the Help tab in the bottom right panel.

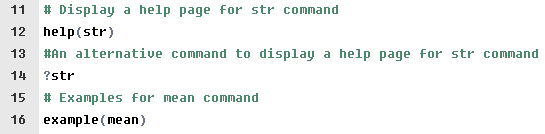
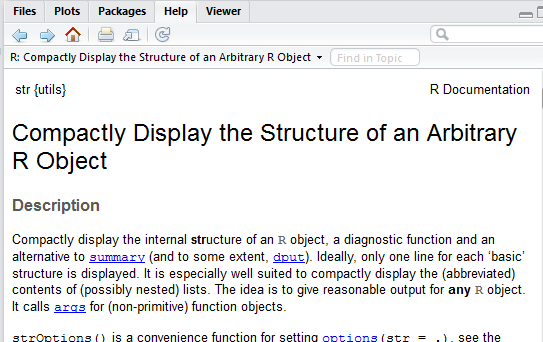


Figure 10: Lines 11-16 in the Exploring iris data script

Run help(str) command on line 12 to view the help page for str command in Figure 11.



Scroll to read the help page content, which includes:

* command description
* command usage
* input parameter information
* links to related help pages
* useful code snippets

Figure 11: Help Page for Str Command

?str command on line 14 is an alternative way to display the help page.

Use the example command to see examples on usage of the specified command. For instance, the command example(mean) on line 16 returns examples for the mean command in Figure 12.

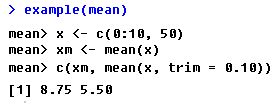


Figure 12: Mean Command Usage Examples

# E:\DBST667 TA\Fall 2015\IrisMeasures.PNGExploring Iris Dataset

The iris dataset contains the flower sepal and petal measurements, including width and length, collected by Fisher’s. These four measurements distinguish three species of flowers - Iris setosa, Iris virginica, and Iris versicolor. The data was collected on 50 flowers of each type. Hence, the dataset contains 150 rows, where each row holds information for one flower.

## Iris Data Help Page

Since iris data is one of the pre-installed datasets, we may run help(iris) command on line 19 to read the help page about the data.

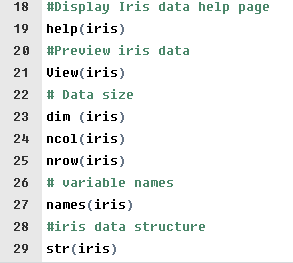


Figure 13: Lines 18-19 in Exploring Iris data script

Figure 14 shows partial content of the help page for the iris data, including the dataset description and dataset versions under the Usage heading. Scroll down the page to read about the dataset variables under the Format heading, and to check the dataset source. The help page also includes useful R code snippets.

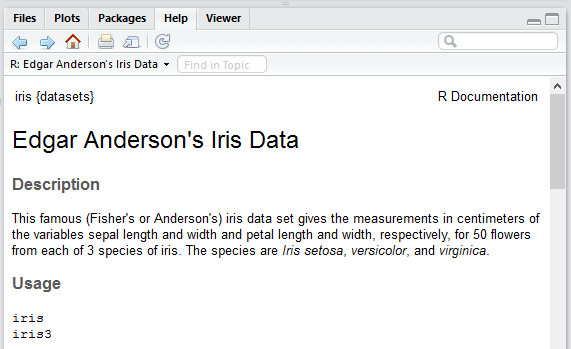


Figure 14: Help Page for Iris Data

## Previewing Iris Data and Exploring its Structure

Figure 15 shows lines 20-53 from the Exploring Iris data script.

Run the View(iris) command on line 21 to preview the iris dataset in a tabular view. (Note that the first letter in the View command is in upper case)

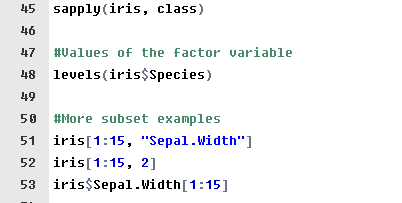


Figure 15: Line 18-53 in Exploring iris data script

Figure 16 shows the first 8 data rows in iris dataset in a tabular format. Each row is an information on one flower. The figure also shows:

* The dataset name in the window title.
* The column headings in the first row are the variable names.
* The values in the first column are the data row labels.
* The heading of the first column is blank.
* Starting from the second column, each column corresponds to a variable.
* Starting from the second row, each row is a single iris record.
* The status bar shows the data rows names in the data preview.
* The status bar also shows the total number of rows in the dataset.
* The arrows next to the variable name allow sorting the data by that variable.
* You may preview the data in a separate window.



Variables

Total number of data rows

Data Rows – each row is an iris record

Dataset name

Heading of the first column is blank

Data rows labels

Previewed rows

Click to view data in a separate window

Click on arrows to sort

Variable Names

Figure 16: Iris Dataset in a Tabular View

Run dim(iris) command on line 23 to display the dataset dimensionality - number of variables and the number of data rows in the dataset. The command takes the dataset name as a parameter. The command output in Figure 17 shows that the iris data has 150 data rows and 5 variables.



150 Data Rows

5 variables

Figure 17: Iris Dataset Dimensionality

Run ncol(iris) command on line 24 to display the number of variables in the dataset. The command takes the dataframe name as an input.

The output in Figure 18 shows that iris dataset has 5 variables.



Figure 18: Number of Variables

Run the nrow(iris) command on line 25 to display the number of rows in iris data. The command takes the dataframe name as an input.

The output in Figure 19 shows that the iris data contains 150 rows.



Figure 19: Number of Data Rows

Run names(iris) command on line 27 to display the iris variables names. The command takes the dataset name as a parameter and returns the variable names enclosed in double quotes in Figure 20.



Figure 20: Names Command Output

Run str(iris) command on line 29 to display the structure of the iris data. The command takes the dataset name as a parameter.

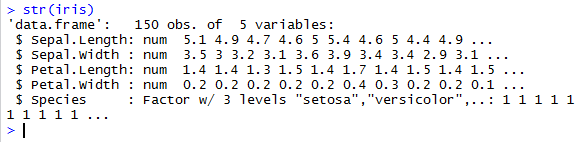
The str(iris) command output in Figure 21 shows that iris is a data frame contains 150 data rows (observations) and 5 variables. The remaining lines show the variables properties. Each line starts with a $ followed by variable name and the variable data type. The variable types in iris dataset are:

**num** - The values of this type of variable are real numbers. For example, the attribute Sepal.length is numeric. The value in the first data row is 5.1. The value in the second data row is 4.9, etc.

**Factor** – The values of this type of variable are discrete pre-defined labels. For each factor variable, the pre-defined discrete values are called levels. In the iris dataset, the last variable Species is a factor variable. The levels are “setosa”, “versicolor”, and ”virginica”. The value 1 in the first data row stands for the first level, or “setosa”.

Number of variables

Number of observations



Species Levels in alphabetical order

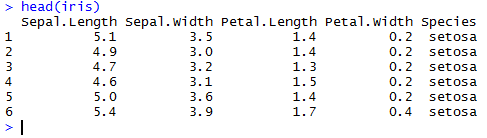
Variable data types

Variable names

Figure 21: str Command Output

Run the head (iris) command on line 31 to display the first six rows in the iris dataset. The command takes the dataset name as a parameter and the number of rows to display as an optional parameter. If the number of rows is omitted, the default is six rows.

Figure 22 shows the output from the head(iris) command. The column headings in the first row are the variable names. The remaining rows are the data rows. The values in the first column are the data row labels.



Variable names

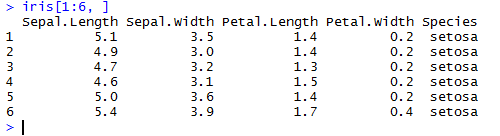
First 6 data rows

Data row labels

Figure 22: Head Command to Display the First 6 Rows

The command iris[1:6, ] on line 33 is an alternative approach to display the first 6 rows. The dataset name is followed by the brackets. We specify the rows to extract. 1:6 means rows 1 through 6. After the comma, we specify the column numbers to extract. If the column numbers are omitted, the output will include all columns.

The output in Figure 23 is identical to the output in Figure 22 above.



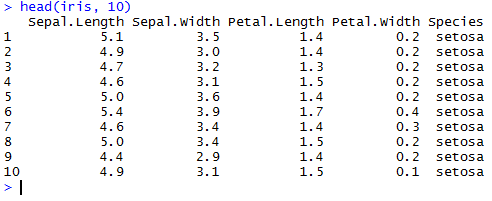
First 6 data rows

Variable names

Figure 23: An Alternative way to Display the First 6 Rows

Run the head (iris, 10) command on line 35 to display the first ten rows in the iris dataset. The command takes the dataset name as a parameter and the number of rows to display as an optional parameter.

Figure 24 shows the returned variable names and the first 10 data rows.



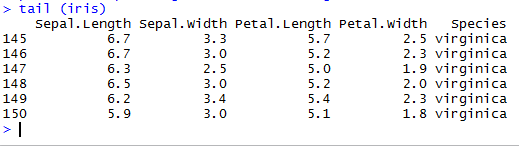
First 10 data rows

Variable names

Figure 24: First 10 Rows

Run the tail(iris) command on line 37 to display the last six rows in the iris dataset. The command takes the dataset name as a required parameter and the number of rows to display as an optional parameter. If the number of rows is omitted, the command returns 6 rows by default

Figure 25 shows the returned variable names and the last 6 rows in the iris data.

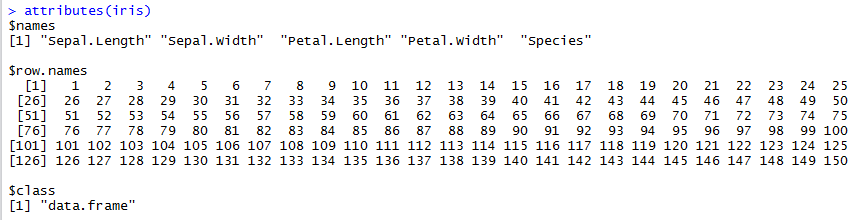


Last 6 Data Rows

Variable names

Figure 25: Last 6 Data Rows

Run the attributes(iris) command on line 40 to display the variable names and the row labels for the iris data in Figure 26.



Data type is the data frame

Data row labels

Variable names

Figure 26: Iris Dataset Attributes

Run the class(iris$Species) command on line 43 to display the data type of the Species variable. To pass a single column as an input to the class command, we suffix the dataset name with a $ followed by the column name.

An output in Figure 27 shows that Species is a factor variable.



Figure 27: Species Datatype

Run the sapply(iris, class) command on line 45 to display the iris data variable names and their corresponding data types in Figure 28. The command takes the data frame name as the first input parameter. The value of the second input parameter is class.

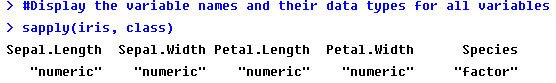


Figure 28: Variable Names and Data Types

To display the values that a categorical variable Species may have, run levels(iris$Species) command on line 48. The command takes the dataframe variable name as an input.

The output in Figure 29 shows the values that variable Species holds.



Figure 29: Levels Command Output

You may display the data subset for a single variable, and there is more than one way to do it. Figure 30 shows that the commands on line 51, 52, and 53 produce identical output.

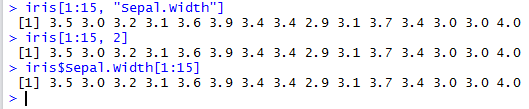


Figure 30: The First 15 values of Sepal.Width

## Descriptive Statistics

We use the summary command to display the descriptive statistics for all variables or for a single variable. To display the statistics for all variables, specify the data frame name as the input parameter to the summary command.

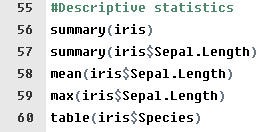
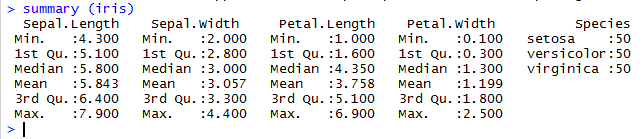


Figure 31: Lines 55-60 in Exploring Iris data script

Run the summary(iris) command on line 56 to display the statistics for all variables in the iris dataset.

The statistics for continuous numeric variables in Figure 32 are the minimum value, maximum value, 1st and 3rd quartile, median, and mean. Variables Sepal.Length. Sepal.Width, Petal.Length, and Petal.Windth are continuous.

The variable Species is a categorical variable, called factor. The variable statistics show the pre-defined discrete values and the number of data rows with each value. The pre-defined species values are setosa, versicolor, and virginica.



Predefined Species Values

Factor Variable

Figure 32: Descriptive Statistics for All Attributes

You may use summary command to display the descriptive statistics for a single variable. Append $ and attribute name to the data frame name. For example, run the summary(iris$Sepal.Length) command on line 57 to display the descriptive statistics for the Sepal.Length attribute in Figure 33.

Variable in the dataset

Dataset name

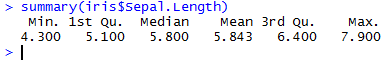


Figure 33: Descriptive statistics for Sepal.length attribute

To display only the mean value of the sepal length variable in Figure 34, run the mean(iris$Sepal.Length) command on line 58.



Figure 34: Sepal.length mean

To display only the maximum value of the sepal length in Figure 35, run the max(iris$Sepal.Length) command on line 59.



Figure 35: Maximum Sepal.length

Run the table(iris$Species) command on line 60 to display the frequency counts for the Species variable. The command takes the variable name as an input.

Figure 36 shows that the dataset has species with the values “setosa”, “versicolor”, and “virginica.” There are 50 species of each type. The sum of the frequency counts is the number of data rows in iris dataset.

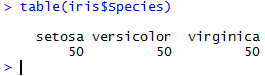


Figure 36: Species Variable Frequency Counts

## Data Visualization

The code on lines 62-74 show the commands to visualize the iris data.

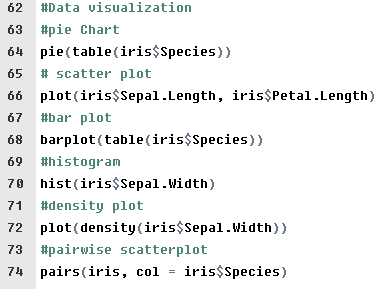
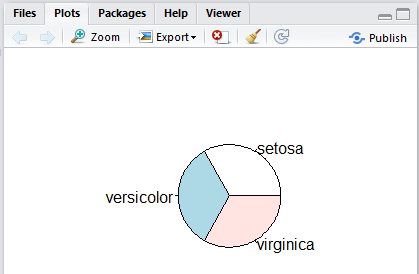


Figure 37: Lines 62-74 in the Exploring iris data script

Run the pie(table(iris$Species)) command on line 64 to build the pie chart for Species frequency counts. The command takes a frequency table as an input.

The plot in Figure 38 will be displayed in the plots tab of the bottom right window. Different color codes represent different species. The slice size indicates the proportion of data rows that have the corresponding species value.



Click to delete all plots

Click to delete the current plot

Export menu allows copying the chart to the clipboard, and saving the chart as a PDF or as an image file.

Click Zoom to open the plot in a new window

Figure 38: Species Variable Pie Chart

Run the plot(iris$Sepal.Length, iris$Petal.Length) command to build the scatter plot in Figure 39. The first parameter is the x-axis variable, and the second parameter is the y-axis variable.

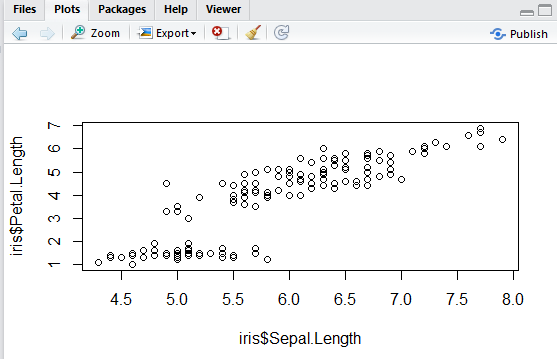
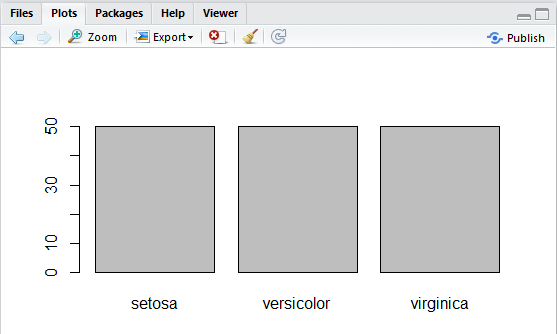


Figure 39: Scatter Plot

Run the barplot(table(iris$Species)) command on line 68 to build the bar plot for the Species variable. The command takes the frequency table for a variable as an input.

The bar plot will be displayed in the Plots tab as shown in Figure 40.



The bar height represents the number of data rows with the corresponding species. Since the dataset has 50 rows for each species, the bar heights are even.

Figure 40: Bar Plot for Species Variable

Run the hist(iris$Sepal.Width) command on line 70 to build the histogram for the Sepal.Width variable. The command takes a variable name as an input.

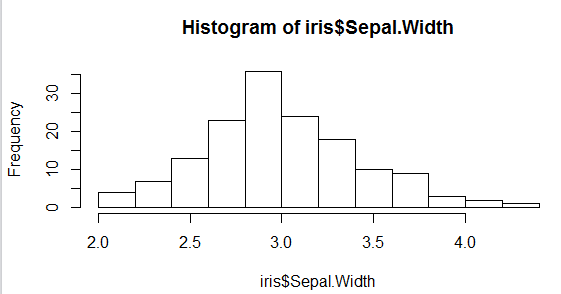


Figure 41: Histogram for Speal.Width Variable

Run the plot(density(iris$Sepal.Width)) command on line 72 to build a density plot for Sepal.width variable.

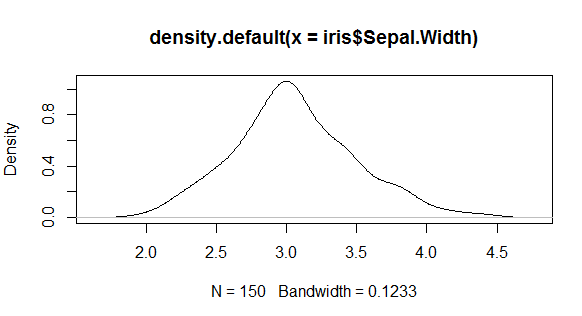


Figure 42: Density Plot for Sepal.Width Variable

Run the pairs function on line 74 to build the pairwise scatterplot in Figure 43. The first input parameter is the data frame name, and the second input parameter specifies the use of a different color for each species value.

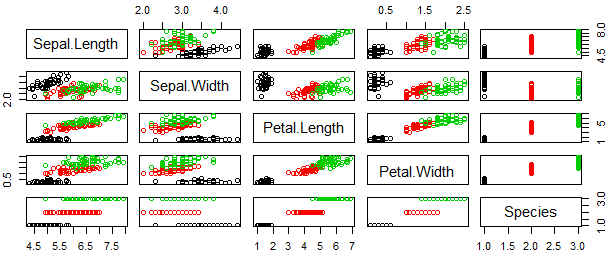


Figure 43: Pairwise Scatter Plot

## Data Aggregation

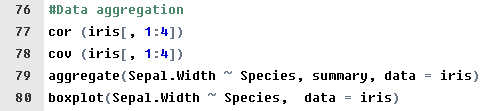


Figure 44: Lines 76-80 in the Exploring Iris Data script

Run the command on line 77 to display the correlations among the numeric variables Sepal.Length, Sepal.Width, Petal.Length, Petal.Width. The command takes the dataframe subset as an input. The row numbers and column numbers to extract are specified in the brackets. Since the row numbers before comma are not specified, the subset includes all rows. The 1:4 after the comma means columns at positions 1, 2, 3, and 4.

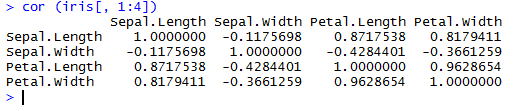


Figure 45: Correlations among Sepal.length, Sepal.Width, Petal.Length, Petal.Width

Run the command on line 78 to display the covariance among the numeric variables Sepal.length, Sepal.Width, Petal.Length, Petal.Width

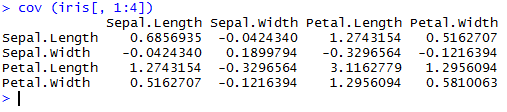


Figure 46: Covarience among Sepal.length, Sepal.Width, Petal.Length, and Petal.Width

The command on line 79 returns the variable aggregate statistics for each value of the factor variable. The first parameter is a numeric variable name followed by the tilde and the variable name used for grouping. The second parameter specifies the statistics to display, and the third parameter is the dataset name.

Figure 47 shows the aggregate statistics for the Sepal.width attribute. The statistics are grouped by the values of the Species variable. The first column is the list of species values. The remaining columns are the statistics. For example, the first value in the second column is a minimum Sepal.Width for setosa species. The second value in the second column is the minimum Sepal.width for versicolor species, etc.

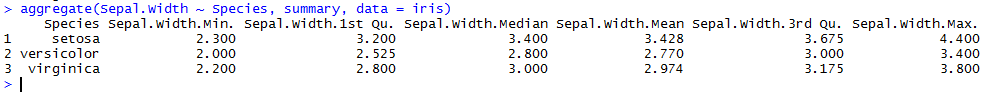


Figure 47: Aggregate Statistics by Species

Run the command on line 80 to build the boxplot for the Sepal.width by Species in Figure 48.

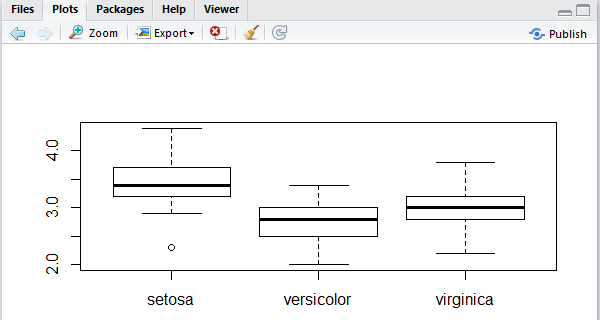


Figure 48: Boxplot for Sepal.width by Species

# How to Create New Script

To create a new R script, select New File from the file menu, and select R Script.

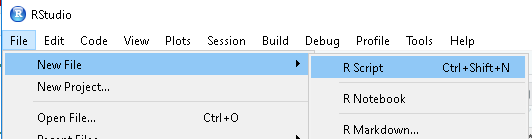


Figure 49: New File ->R Script file

The new file will open in the source editor. Type the code in the editor.

Click on the Save button on the source window tool bar to save the file.

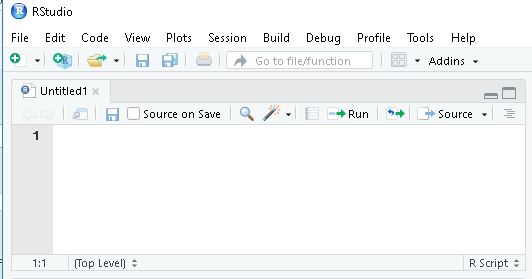


Figure 50: New Script file in the source editor

# Exit R Studio

To exit the program, you may execute the following command in the console window.

q(). You may also choose Quit RStudio from the file menu. You will be prompted to save the workspace.