R Function Summary

Last Updated: May 14, 2023

Contents

Purpose

This file contains commonly used R functions for data analysis, data management, and visualization. All code examples will use iris, a pre-loaded R data set. Type ?iris in the console to learn more information about the dataset.

Data Management

1. factor()

Package: base R

```
#Explanation: Specifies hierarchy among a set of values or a column. The code below sets setosa to the

# Convert the Species column to a factor.
iris$Species <- factor(iris$Species)

# Get the levels of the Species factor
levels(iris$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

```
# Check the class of the Species column
class(iris$Species)
```

```
## [1] "factor"
```

2. for()

Package: base R

```
#Explanation: For loops performs a set of operations on each value in a sequence.

#Loop over the columns of the iris dataset
for (col in names(iris)) {
    # Calculate the mean of the column
    col_mean <- mean(iris[[col]])
    # Print the column name and mean
    cat("Mean of", col, ":", col_mean, "\n")
  }
```

```
## Mean of Sepal.Length : 5.843333
## Mean of Sepal.Width : 3.057333
## Mean of Petal.Length : 3.758
## Mean of Petal.Width : 1.199333
## Mean of Species : NA
```

3. function()

Package: base R

```
#Explanation: Makes a new function that can execute a specific task through a combination of specified

# Define a function that calculates the mean of a numeric column in a data frame
mean_column <- function(df, column) {
  if (!is.numeric(df[[column]])) {
    stop("Column must be numeric")
  }
  return(mean(df[[column]]))
}

# Calculate the mean of the Sepal.Length column using the mean_column() function
sepal_length_mean <- mean_column(iris, "Sepal.Length")
sepal_length_mean
```

```
## [1] 5.843333
```

4. ifelse()

Package: base R

```
#Explanation: If a condition is met, then a action or function is executed. If the condition is not met

# Create a new column called "SpeciesCode" based on the "Species" column
iris$SpeciesCode <- ifelse(iris$Species == "setosa", 1,
                           ifelse(iris$Species == "versicolor", 2, 3))

# Print the first 10 rows of the iris dataset
head(iris, 10)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species SpeciesCode
## 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
## 7           4.6         3.4          1.4         0.3  setosa           1
## 8           5.0         3.4          1.5         0.2  setosa           1
## 9           4.4         2.9          1.4         0.2  setosa           1
## 10          4.9         3.1          1.5         0.1  setosa           1
```

5. is.na()

Package: base R

```
#Explanation: Identifies if value is missing (NA) or not

#Count all missing values in iris
sum(is.na(iris))
```

```
## [1] 0
```

```
#Count all non-missing values in iris
sum(!is.na(iris)) # ! = not
```

```
## [1] 900
```

6. lapply()

Package: base R

```
#Explanation: Applies a function on each column or value of a dataset and returns a list

# Apply the `mean()` function to each numeric column of the iris dataset
means <- lapply(iris[,1:4], mean)

# Print the resulting list of means
print(means)
```

```
## $Sepal.Length
## [1] 5.843333
##
## $Sepal.Width
## [1] 3.057333
##
## $Petal.Length
## [1] 3.758
##
## $Petal.Width
## [1] 1.199333
```

7. merge()

Package: base R

```
#Explanation: Combines two dataframes which have one column with identical values

#Normally you merge two different datasets, but we will merge iris with itself
merged_iris <- merge(iris, iris, #names of two datasets
                     by = c("Species"), #shared column between two datasets
                     suffixes = c("_1", "_2")) #adding suffixes to distinguish orginal values

head(merged_iris) #see the first 6 rows
```

```
##   Species Sepal.Length_1 Sepal.Width_1 Petal.Length_1 Petal.Width_1
## 1  setosa            5.1           3.5            1.4           0.2
## 2  setosa            5.1           3.5            1.4           0.2
## 3  setosa            5.1           3.5            1.4           0.2
## 4  setosa            5.1           3.5            1.4           0.2
## 5  setosa            5.1           3.5            1.4           0.2
## 6  setosa            5.1           3.5            1.4           0.2
##   SpeciesCode_1 Sepal.Length_2 Sepal.Width_2 Petal.Length_2 Petal.Width_2
## 1             1            5.1           3.5            1.4           0.2
## 2             1            4.9           3.0            1.4           0.2
## 3             1            4.7           3.2            1.3           0.2
## 4             1            4.6           3.1            1.5           0.2
## 5             1            5.0           3.6            1.4           0.2
## 6             1            5.4           3.9            1.7           0.4
##   SpeciesCode_2
## 1             1
## 2             1
## 3             1
## 4             1
## 5             1
## 6             1
```

8. read.csv()

Package: base R

```
#Explanation: Loads .csv files into R

#iris2 <- read.csv("name_of_file.csv")
#make sure your working directory is set to your file location.
```

9. rename()

Package: dplyr

```
#Explanation: Changes column names
library(dplyr) #part of tidyverse
```

```r
iris_renamed <- iris %>% rename(SL = Sepal.Length, #Changed Sepal.Length to SL
                                SW = Sepal.Width,  #Changed Sepal.Width to SW
                                PL = Petal.Length, #Changed Petal.Length to PL
                                PW = Petal.Width,  #Changed Petal.Width to PW
                                Class = Species)   #Changed Species to Class
iris_renamed
```

```
##      SL  SW  PL  PW      Class SpeciesCode
## 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
## 7   4.6 3.4 1.4 0.3     setosa           1
## 8   5.0 3.4 1.5 0.2     setosa           1
## 9   4.4 2.9 1.4 0.2     setosa           1
## 10  4.9 3.1 1.5 0.1     setosa           1
## 11  5.4 3.7 1.5 0.2     setosa           1
## 12  4.8 3.4 1.6 0.2     setosa           1
## 13  4.8 3.0 1.4 0.1     setosa           1
## 14  4.3 3.0 1.1 0.1     setosa           1
## 15  5.8 4.0 1.2 0.2     setosa           1
## 16  5.7 4.4 1.5 0.4     setosa           1
## 17  5.4 3.9 1.3 0.4     setosa           1
## 18  5.1 3.5 1.4 0.3     setosa           1
## 19  5.7 3.8 1.7 0.3     setosa           1
## 20  5.1 3.8 1.5 0.3     setosa           1
## 21  5.4 3.4 1.7 0.2     setosa           1
## 22  5.1 3.7 1.5 0.4     setosa           1
## 23  4.6 3.6 1.0 0.2     setosa           1
## 24  5.1 3.3 1.7 0.5     setosa           1
## 25  4.8 3.4 1.9 0.2     setosa           1
## 26  5.0 3.0 1.6 0.2     setosa           1
## 27  5.0 3.4 1.6 0.4     setosa           1
## 28  5.2 3.5 1.5 0.2     setosa           1
## 29  5.2 3.4 1.4 0.2     setosa           1
## 30  4.7 3.2 1.6 0.2     setosa           1
## 31  4.8 3.1 1.6 0.2     setosa           1
## 32  5.4 3.4 1.5 0.4     setosa           1
## 33  5.2 4.1 1.5 0.1     setosa           1
## 34  5.5 4.2 1.4 0.2     setosa           1
## 35  4.9 3.1 1.5 0.2     setosa           1
## 36  5.0 3.2 1.2 0.2     setosa           1
## 37  5.5 3.5 1.3 0.2     setosa           1
## 38  4.9 3.6 1.4 0.1     setosa           1
## 39  4.4 3.0 1.3 0.2     setosa           1
## 40  5.1 3.4 1.5 0.2     setosa           1
## 41  5.0 3.5 1.3 0.3     setosa           1
## 42  4.5 2.3 1.3 0.3     setosa           1
## 43  4.4 3.2 1.3 0.2     setosa           1
## 44  5.0 3.5 1.6 0.6     setosa           1
## 45  5.1 3.8 1.9 0.4     setosa           1
## 46  4.8 3.0 1.4 0.3     setosa           1
## 47  5.1 3.8 1.6 0.2     setosa           1
## 48  4.6 3.2 1.4 0.2     setosa           1
## 49  5.3 3.7 1.5 0.2     setosa           1
## 50  5.0 3.3 1.4 0.2     setosa           1
## 51  7.0 3.2 4.7 1.4 versicolor           2
## 52  6.4 3.2 4.5 1.5 versicolor           2
## 53  6.9 3.1 4.9 1.5 versicolor           2
## 54  5.5 2.3 4.0 1.3 versicolor           2
## 55  6.5 2.8 4.6 1.5 versicolor           2
## 56  5.7 2.8 4.5 1.3 versicolor           2
## 57  6.3 3.3 4.7 1.6 versicolor           2
## 58  4.9 2.4 3.3 1.0 versicolor           2
## 59  6.6 2.9 4.6 1.3 versicolor           2
## 60  5.2 2.7 3.9 1.4 versicolor           2
## 61  5.0 2.0 3.5 1.0 versicolor           2
## 62  5.9 3.0 4.2 1.5 versicolor           2
## 63  6.0 2.2 4.0 1.0 versicolor           2
## 64  6.1 2.9 4.7 1.4 versicolor           2
## 65  5.6 2.9 3.6 1.3 versicolor           2
## 66  6.7 3.1 4.4 1.4 versicolor           2
## 67  5.6 3.0 4.5 1.5 versicolor           2
## 68  5.8 2.7 4.1 1.0 versicolor           2
## 69  6.2 2.2 4.5 1.5 versicolor           2
## 70  5.6 2.5 3.9 1.1 versicolor           2
## 71  5.9 3.2 4.8 1.8 versicolor           2
## 72  6.1 2.8 4.0 1.3 versicolor           2
## 73  6.3 2.5 4.9 1.5 versicolor           2
## 74  6.1 2.8 4.7 1.2 versicolor           2
## 75  6.4 2.9 4.3 1.3 versicolor           2
## 76  6.6 3.0 4.4 1.4 versicolor           2
## 77  6.8 2.8 4.8 1.4 versicolor           2
## 78  6.7 3.0 5.0 1.7 versicolor           2
## 79  6.0 2.9 4.5 1.5 versicolor           2
## 80  5.7 2.6 3.5 1.0 versicolor           2
## 81  5.5 2.4 3.8 1.1 versicolor           2
## 82  5.5 2.4 3.7 1.0 versicolor           2
## 83  5.8 2.7 3.9 1.2 versicolor           2
## 84  6.0 2.7 5.1 1.6 versicolor           2
## 85  5.4 3.0 4.5 1.5 versicolor           2
## 86  6.0 3.4 4.5 1.6 versicolor           2
## 87  6.7 3.1 4.7 1.5 versicolor           2
## 88  6.3 2.3 4.4 1.3 versicolor           2
## 89  5.6 3.0 4.1 1.3 versicolor           2
## 90  5.5 2.5 4.0 1.3 versicolor           2
## 91  5.5 2.6 4.4 1.2 versicolor           2
## 92  6.1 3.0 4.6 1.4 versicolor           2
## 93  5.8 2.6 4.0 1.2 versicolor           2
## 94  5.0 2.3 3.3 1.0 versicolor           2
## 95  5.6 2.7 4.2 1.3 versicolor           2
## 96  5.7 3.0 4.2 1.2 versicolor           2
## 97  5.7 2.9 4.2 1.3 versicolor           2
## 98  6.2 2.9 4.3 1.3 versicolor           2
## 99  5.1 2.5 3.0 1.1 versicolor           2
## 100 5.7 2.8 4.1 1.3 versicolor           2
## 101 6.3 3.3 6.0 2.5  virginica           3
## 102 5.8 2.7 5.1 1.9  virginica           3
## 103 7.1 3.0 5.9 2.1  virginica           3
## 104 6.3 2.9 5.6 1.8  virginica           3
## 105 6.5 3.0 5.8 2.2  virginica           3
## 106 7.6 3.0 6.6 2.1  virginica           3
## 107 4.9 2.5 4.5 1.7  virginica           3
## 108 7.3 2.9 6.3 1.8  virginica           3
## 109 6.7 2.5 5.8 1.8  virginica           3
## 110 7.2 3.6 6.1 2.5  virginica           3
## 111 6.5 3.2 5.1 2.0  virginica           3
## 112 6.4 2.7 5.3 1.9  virginica           3
## 113 6.8 3.0 5.5 2.1  virginica           3
## 114 5.7 2.5 5.0 2.0  virginica           3
## 115 5.8 2.8 5.1 2.4  virginica           3
## 116 6.4 3.2 5.3 2.3  virginica           3
## 117 6.5 3.0 5.5 1.8  virginica           3
## 118 7.7 3.8 6.7 2.2  virginica           3
## 119 7.7 2.6 6.9 2.3  virginica           3
## 120 6.0 2.2 5.0 1.5  virginica           3
## 121 6.9 3.2 5.7 2.3  virginica           3
## 122 5.6 2.8 4.9 2.0  virginica           3
## 123 7.7 2.8 6.7 2.0  virginica           3
## 124 6.3 2.7 4.9 1.8  virginica           3
## 125 6.7 3.3 5.7 2.1  virginica           3
## 126 7.2 3.2 6.0 1.8  virginica           3
## 127 6.2 2.8 4.8 1.8  virginica           3
## 128 6.1 3.0 4.9 1.8  virginica           3
## 129 6.4 2.8 5.6 2.1  virginica           3
## 130 7.2 3.0 5.8 1.6  virginica           3
## 131 7.4 2.8 6.1 1.9  virginica           3
## 132 7.9 3.8 6.4 2.0  virginica           3
## 133 6.4 2.8 5.6 2.2  virginica           3
## 134 6.3 2.8 5.1 1.5  virginica           3
## 135 6.1 2.6 5.6 1.4  virginica           3
## 136 7.7 3.0 6.1 2.3  virginica           3
## 137 6.3 3.4 5.6 2.4  virginica           3
## 138 6.4 3.1 5.5 1.8  virginica           3
## 139 6.0 3.0 4.8 1.8  virginica           3
## 140 6.9 3.1 5.4 2.1  virginica           3
## 141 6.7 3.1 5.6 2.4  virginica           3
## 142 6.9 3.1 5.1 2.3  virginica           3
## 143 5.8 2.7 5.1 1.9  virginica           3
## 144 6.8 3.2 5.9 2.3  virginica           3
## 145 6.7 3.3 5.7 2.5  virginica           3
## 146 6.7 3.0 5.2 2.3  virginica           3
## 147 6.3 2.5 5.0 1.9  virginica           3
## 148 6.5 3.0 5.2 2.0  virginica           3
## 149 6.2 3.4 5.4 2.3  virginica           3
## 150 5.9 3.0 5.1 1.8  virginica           3
```

10. sapply()

Package: base R

```
#Explanation: Applies a function on each column or value of a dataset and returns a vector or matrix

# Apply the mean function to each column of the dataset
column_means <- sapply(X = iris, FUN = mean, na.rm = TRUE)

# View the column means
column_means #Species is categorical, so NA is returned
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species  SpeciesCode
##     5.843333     3.057333     3.758000     1.199333           NA     2.000000
```

11. writeLines

Package: base R

```
#Explanation: Writes one or more character strings to text. Useful in working with strings.

# Write the first 5 rows of the dataset to a text file
writeLines(colnames(iris))
```

```
## Sepal.Length
## Sepal.Width
## Petal.Length
## Petal.Width
## Species
## SpeciesCode
```

12. read.xlsx ()

Package: readxl

```
#Explanation: Loads .xlsx files into R
library(readxl)

# Read the Excel file
# example <- read.xlsx("name_of_your_file.xlsx",
#                      sheetIndex = 1, #will read only the first sheet
#                      startRow = 2, #will start reading the data at the 2nd row
#                      endRow = 100,  #will stop reading the data the 100th row
#                      colIndex = 1:5) #will only read columns 1-5
```

13. write.xlsx

Package: readxl

```
#Explanation: Saves dataframe as an .xlsx file type
# write.xlsx(df, #name of object
#             file = "example.xlsx", #name that of file being created
#             sheetName = "Sheet1",  #name that of sheet being created
#             row.names = FALSE)     #not including row names
```

14. str_c()

Package: stringr

```
#Explanation: Combines multiple values (characters, special characters, and/or numbers) together

library(stringr)

# Concatenate the species and petal width columns of the iris data frame
iris$Species_PetalWidth <- str_c(iris$Species, " - ", iris$Petal.Width)

# View the first few rows of the modified iris data frame
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species SpeciesCode
## 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
##   Species_PetalWidth
## 1       setosa - 0.2
## 2       setosa - 0.2
## 3       setosa - 0.2
## 4       setosa - 0.2
## 5       setosa - 0.2
## 6       setosa - 0.4
```

15. str_match()

Package: stringr

```
#Explanation: Locates and stores all occurrences of specificed characters, special characters, or numbe
library(stringr)

# Extract the petal length from the iris data frame using str_match
iris$Petal.Length.Extracted <- str_match(iris$Petal.Length, #variable
                                         "\\d+\\.\\d+") #searching for a digit (d), followed by a perio

# View the first few rows of the modified iris data frame
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species SpeciesCode
## 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
##   Species_PetalWidth Petal.Length.Extracted
## 1       setosa - 0.2                    1.4
## 2       setosa - 0.2                    1.4
## 3       setosa - 0.2                    1.3
## 4       setosa - 0.2                    1.5
## 5       setosa - 0.2                    1.4
## 6       setosa - 0.4                    1.7
```

16. str_replace ()

Package: stringr

```
#Explanation: Replaces characters, special characters, or numbers in a column (or cell) with specified
library(stringr)

# Replace the species names in the iris data frame with shorter names
iris$Species <- str_replace(iris$Species, #variable of interest
                            "versicolor", #character to search for
                            "vers")       #replacement text

# View the change name
table(iris$Species)
```

```
##
##    setosa      vers virginica
##        50        50        50
```

17. str_view_all()

Package: stringr

```
#Explanation: Shows all occurances of specified characters, special characters, or numbers within a cel
library(stringr)

# Use str_view_all to visualize the matches of the "setosa" pattern in the Species column of the iris d
str_view_all(iris$Species, "setosa")
```

```
##  [1] | <setosa>
##  [2] | <setosa>
##  [3] | <setosa>
##  [4] | <setosa>
##  [5] | <setosa>
##  [6] | <setosa>
##  [7] | <setosa>
##  [8] | <setosa>
##  [9] | <setosa>
## [10] | <setosa>
## [11] | <setosa>
## [12] | <setosa>
## [13] | <setosa>
## [14] | <setosa>
## [15] | <setosa>
## [16] | <setosa>
## [17] | <setosa>
## [18] | <setosa>
## [19] | <setosa>
## [20] | <setosa>
## ... and 130 more
```

18. arrange()

Package: dplyr

```
#Explanation: Sorts rows in a dataframe based on one or more columns
library(tidyverse) #dplyr is part of tidyverse

# Sort the iris data frame by Petal.Length in ascending order
iris_sorted <- arrange(iris, Petal.Length)

# View the first few rows of the sorted data frame
head(iris_sorted)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species SpeciesCode
## 1          4.6         3.6          1.0         0.2  setosa           1
## 2          4.3         3.0          1.1         0.1  setosa           1
## 3          5.8         4.0          1.2         0.2  setosa           1
## 4          5.0         3.2          1.2         0.2  setosa           1
## 5          4.7         3.2          1.3         0.2  setosa           1
## 6          5.4         3.9          1.3         0.4  setosa           1
##   Species_PetalWidth Petal.Length.Extracted
## 1       setosa - 0.2                   <NA>
## 2       setosa - 0.1                    1.1
## 3       setosa - 0.2                    1.2
## 4       setosa - 0.2                    1.2
## 5       setosa - 0.2                    1.3
## 6       setosa - 0.4                    1.3
```

19. bind_cols

Package: dplyr

```
#Explanation: Adds columns of a dataframe to another dataframe
library(tidyverse) #dplyr is part of tidyverse

# Create a new data frame with two columns
new_data <- data.frame(Sepal.Width = runif(150, 2.5, 4),
                       Sepal.Length = runif(150, 4, 7))
```

```
# Combine the iris and new_data data frames by column
iris_combined <- bind_cols(iris, new_data)

# View the first few rows of the combined data frame
head(iris_combined)
```

```
##   Sepal.Length...1 Sepal.Width...2 Petal.Length Petal.Width Species SpeciesCode
## 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
##   Species_PetalWidth Petal.Length.Extracted Sepal.Width...9 Sepal.Length...10
## 1       setosa - 0.2                    1.4        3.964920          4.387710
## 2       setosa - 0.2                    1.4        2.649119          5.074333
## 3       setosa - 0.2                    1.3        3.941582          6.982639
## 4       setosa - 0.2                    1.5        2.862421          5.980577
## 5       setosa - 0.2                    1.4        3.448873          6.054903
## 6       setosa - 0.4                    1.7        3.202657          4.240079
```

20. bind_rows

Package: dplyr

```
#Explanation: Adds rows of a dataframe to another dataframe
library(tidyverse) #dplyr is part of tidyverse

# Create a new data frame with two rows
new_data <- data.frame(Sepal.Width = c(3.5, 2.8),
                       Sepal.Length = c(6.2, 5.1),
                       Petal.Length = c(4.3, 1.5),
                       Petal.Width = c(1.3, 0.3),
                       Species = c("versicolor", "setosa"))

# Combine the iris and new_data data frames by row
iris_combined <- bind_rows(iris, new_data)

# View the last few rows of the combined data frame
tail(iris_combined)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species SpeciesCode
## 147          6.3         2.5          5.0         1.9  virginica           3
## 148          6.5         3.0          5.2         2.0  virginica           3
## 149          6.2         3.4          5.4         2.3  virginica           3
## 150          5.9         3.0          5.1         1.8  virginica           3
## 151          6.2         3.5          4.3         1.3 versicolor          NA
## 152          5.1         2.8          1.5         0.3     setosa          NA
##     Species_PetalWidth Petal.Length.Extracted
## 147    virginica - 1.9                   <NA>
## 148      virginica - 2                    5.2
## 149    virginica - 2.3                    5.4
## 150    virginica - 1.8                    5.1
## 151               <NA>                   <NA>
## 152               <NA>                   <NA>
```

21. filter()

Package: dplyr

```
#Explanation: Searches for values that fit one or more user-defined conditions
library(tidyverse) #dplyr is part of tidyverse

# Extract all rows where Species is "setosa"
setosa <- filter(iris, #df
                 Species == #variable
                   "setosa") #condition to search for

# Extract all rows where Sepal.Length is greater than 6.0
long_sepal <- filter(iris, Sepal.Length > 6.0)

# Extract all rows where Petal.Length is between 4.5 and 5.0
long_petal <- filter(iris, Petal.Length >= 4.5, Petal.Length <= 5.0)
```

22. mutate()

Package: dplyr

```
#Explanation: Creates a new column
library(tidyverse) #dplyr is part of tidyverse

# Add a new column that is the product of Sepal.Length and Sepal.Width
iris <- mutate(iris, #dataframe
               sepal_area = #name of new column
                 Sepal.Length * Sepal.Width) #formula to create values for new column

# Add a new column that is the ratio of Petal.Length to Petal.Width
iris <- mutate(iris, petal_ratio = Petal.Length / Petal.Width)

# Add a new column that is the average of Sepal.Length and Petal.Length
iris <- mutate(iris,avg_length = (Sepal.Length + Petal.Length) / 2)
```

23. select()

Package: dplyr

```
#Explanation: Selects column(s) within a dataframe. Helpful when making a new dataframe with less colum
library(tidyverse) #dplyr is part of tidyverse

# Select the columns Sepal.Length, Sepal.Width, and Species
iris_select <- select(iris, #dataframe
                      Sepal.Length, Sepal.Width, Species) #variables
```

```
# Select all columns except Sepal.Length and Sepal.Width
iris_select <- iris %>% select(-Sepal.Length, -Sepal.Width) #with pipes

# Select columns Petal.Length and Petal.Width, and all columns starting with Sepal
iris_select <- select(iris, Petal.Length, Petal.Width, starts_with("Sepal"))
```

Descriptive Analysis

24. lm()

Package: base R

```
#Explanation: Regresses one or more columns onto another column.

# Fit a linear regression model of Petal.Length on Sepal.Length and Sepal.Width
model <- lm(Petal.Length ~ #dependent variables
              Sepal.Length + Sepal.Width, #independent variables
            data = iris) #data

# Print the summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = Petal.Length ~ Sepal.Length + Sepal.Width, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25582 -0.46922 -0.05741  0.45530  1.75599
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.52476    0.56344  -4.481 1.48e-05 ***
## Sepal.Length  1.77559    0.06441  27.569  < 2e-16 ***
## Sepal.Width  -1.33862    0.12236 -10.940  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.6465 on 147 degrees of freedom
## Multiple R-squared:  0.8677, Adjusted R-squared:  0.8659
## F-statistic:   482 on 2 and 147 DF,  p-value: < 2.2e-16
```

25. prop.table()

Package: base R

```
#Explanation: Converts a frequency or contignecy table into proportions

# Compute the proportions of each species in the iris data set
prop.table(table(iris$Species))
```

```
##
##    setosa      vers virginica
## 0.3333333 0.3333333 0.3333333
```

```
# Compute the proportions of each species by combining the Species and Petal.Width variables
prop.table(table(iris$Species, iris$Petal.Width), margin = 1)
```

```
##
##              0.1  0.2  0.3  0.4  0.5  0.6    1  1.1  1.2  1.3  1.4  1.5  1.6
##   setosa    0.10 0.58 0.14 0.14 0.02 0.02 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##   vers      0.00 0.00 0.00 0.00 0.00 0.00 0.14 0.06 0.10 0.26 0.14 0.20 0.06
##   virginica 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.02 0.04 0.02
##
##              1.7  1.8  1.9    2  2.1  2.2  2.3  2.4  2.5
##   setosa    0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##   vers      0.02 0.02 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##   virginica 0.02 0.22 0.10 0.12 0.12 0.06 0.16 0.06 0.06
```

26. Summary()

Package: base R

```
#Explanation: Lists the quarantines, median, mean, and range of one or more columns

# Obtain a summary of the dataset
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           SpeciesCode Species_PetalWidth Petal.Length.Extracted.V1
##  Length:150          Min.   :1    Length:150         Length:150
##  Class :character   1st Qu.:1    Class :character   Class :character
##  Mode  :character   Median :2    Mode  :character   Mode  :character
##                     Mean   :2
##                     3rd Qu.:3
##                     Max.   :3
##    sepal_area      petal_ratio       avg_length
##  Min.   :10.00   Min.   : 2.125   Min.   :2.700
##  1st Qu.:15.66   1st Qu.: 2.802   1st Qu.:3.362
##  Median :17.66   Median : 3.300   Median :5.050
##  Mean   :17.82   Mean   : 4.311   Mean   :4.801
##  3rd Qu.:20.32   3rd Qu.: 4.667   3rd Qu.:5.800
##  Max.   :30.02   Max.   :15.000   Max.   :7.300
```

27. table()

Package: base R

```
#Explanation: Creates a table that shows the amount of times a value is repeated in one or more columns

#Create table showing the count of 3 Species
table(iris$Species)
```

```
## 
##    setosa      vers virginica 
##        50        50        50 
```

28. xtabs()

Package: base R

```
#Explanation: Creates a contingency table which shows the count for two or more variables

# Create a contingency table of Species and Sepal.Width
xtabs(~ Species + Sepal.Width, #two variables
      data = iris) #dataset
```

```
##             Sepal.Width
## Species      2 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9  3 3.1 3.2 3.3 3.4 3.5 3.6 3.7
##   setosa     0   0   1   0   0   0   0   0   1  6   4   5   2   9   6   3   3
##   vers       1   2   3   3   4   3   5   6   7  8   3   3   1   1   0   0   0
##   virginica  0   1   0   0   4   2   4   8   2 12   4   5   3   2   0   1   0
##             Sepal.Width
## Species     3.8 3.9  4 4.1 4.2 4.4
##   setosa      4   2  1   1   1   1
##   vers        0   0  0   0   0   0
##   virginica   2   0  0   0   0   0
```

29. aggregate()

Package: base R

```
#Explanation: Groups data by one or more variables and then applies a function (i.e. mean)

# Summarize Sepal.Length by Species
aggregate(Sepal.Length ~ Species, #variables
          data = iris, #dataset
          FUN = mean) #function
```

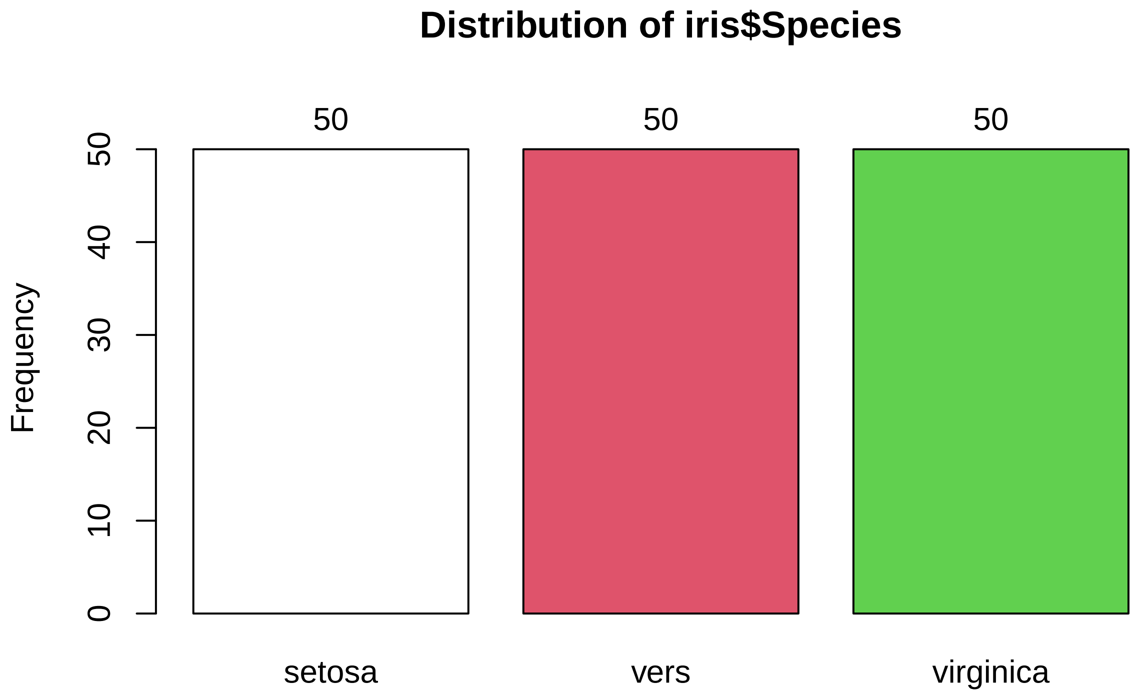
```
##     Species Sepal.Length
## 1    setosa        5.006
## 2      vers        5.936
## 3 virginica        6.588
```

30. tab1()

Package: epiDisplay

```
#Explanation: Creates a table that shows the frequnecy, percent, and cumulative percent for categorical
library(epiDisplay)

tab1(iris$Species)
```



```
## iris$Species :
##           Frequency Percent Cum. percent
## setosa           50    33.3         33.3
## vers             50    33.3         66.7
## virginica        50    33.3        100.0
##   Total         150   100.0        100.0
```

31. tbl_summary()

Package: gtsummary

```
#Explanation: Creates presentation style table that shows the subgroup count, percentages, and total co
library(gtsummary)

#Preloaded dataset 'mtcars' is used since function is better illustrated with this data.

tbl_summary(mtcars, include = c("mpg", "cyl", "disp"))
```

Characteristic	N = 32
mpg	19.2 (15.4, 22.8)
cyl	
4	11 (34%)
6	7 (22%)
8	14 (44%)
disp	196 (121, 326)

32. count ()

Package: dplyr

```
#Explanation: Counts the number of rows in a dataframe or that number of rows that meet specified crite
library(dplyr) #dplyr is part of tidyverse

#Count total observations in dataset
iris %>%
  count()
```

```
##     n
## 1 150
```

```
#Count observations by subgroup Species
iris %>%
  group_by(Species) %>%
  count()
```

```
## # A tibble: 3 x 2
## # Groups:   Species [3]
##   Species       n
##   <chr>     <int>
## 1 setosa       50
## 2 vers         50
## 3 virginica    50
```

33. group_by () & summarize ()

Package: dplyr

```
#Explanation: Group_by subgroups a variable. Summarize can apply a variety of statistical functions for
library(dplyr) #dplyr is part of tidyverse

iris %>%
  group_by(Species) %>%
  summarize(mean_sepal_length = mean(Sepal.Length),
            mean_sepal_width = mean(Sepal.Width),
            mean_petal_length = mean(Petal.Length),
            mean_petal_width = mean(Petal.Width))
```

```
## # A tibble: 3 x 5
##   Species  mean_sepal_length mean_sepal_width mean_petal_length mean_petal_width
##   <chr>                <dbl>            <dbl>             <dbl>            <dbl>
## 1 setosa                5.01             3.43              1.46            0.246
## 2 vers                  5.94             2.77              4.26            1.33
## 3 virgini~              6.59             2.97              5.55            2.03
```

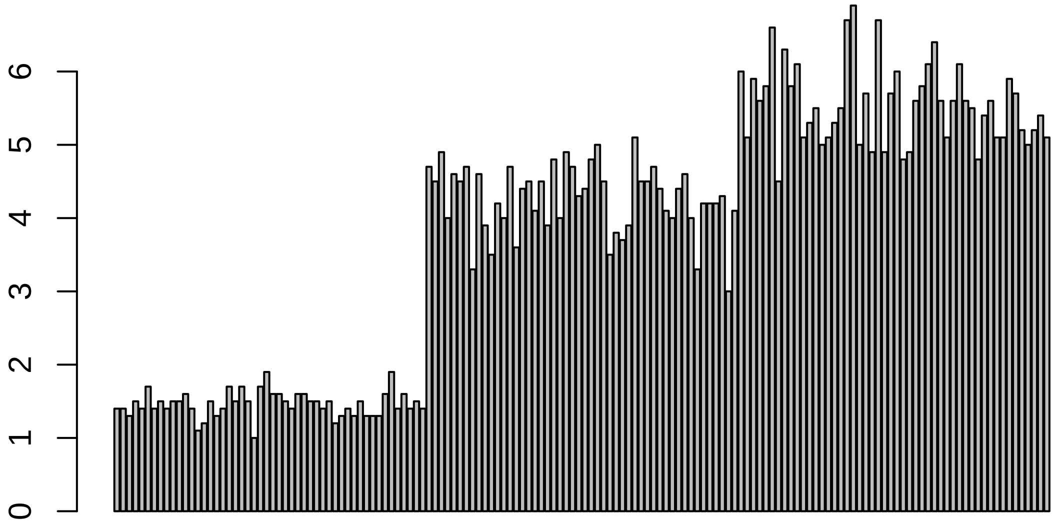
Visualization

34. png()

Package: base R

```
#Explanation: Saves plots file type .png

#create simple bar plot of Petal.Length
barplot(iris$Petal.Length)
```



```
png("iris_boxplot.png")
dev.off #Close PNG
```

```
## function (which = dev.cur())
## {
```

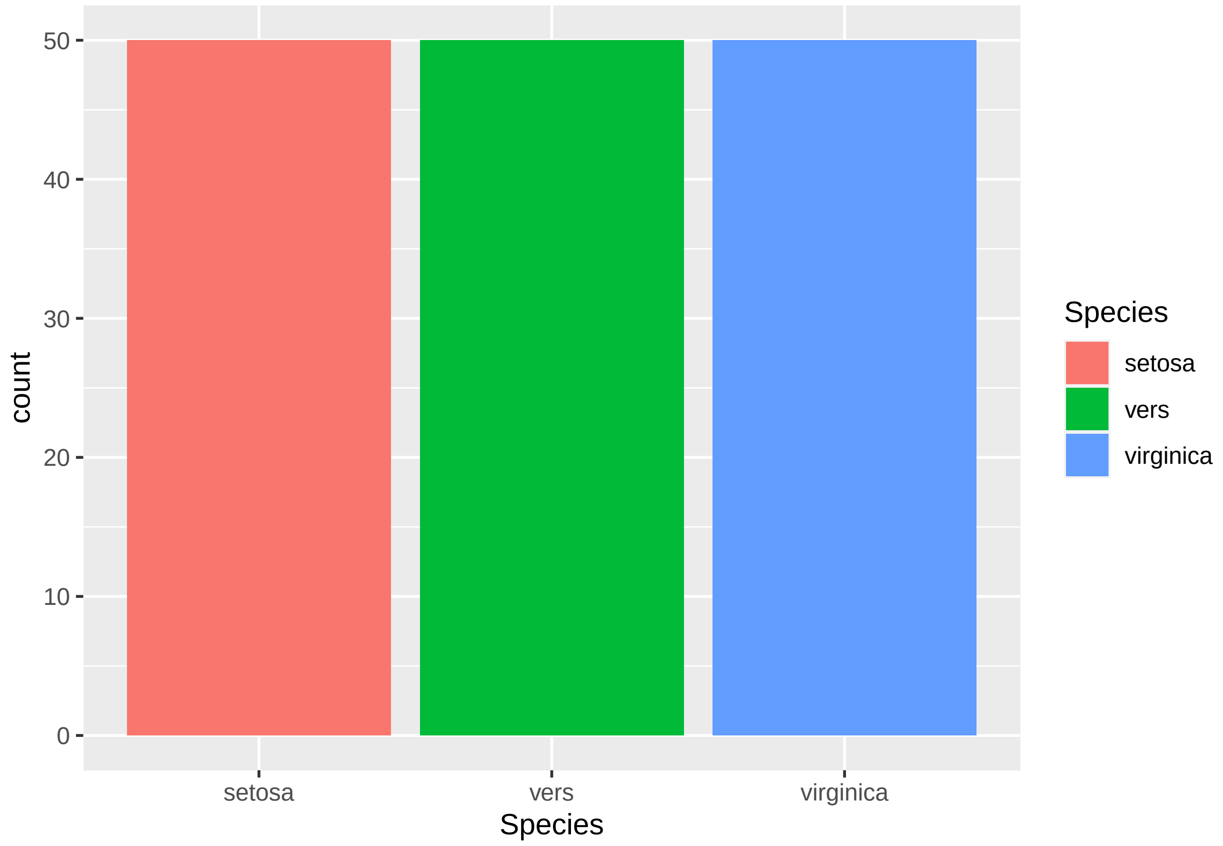
```
##     if (which == 1)
##         stop("cannot shut down device 1 (the null device)")
##     .External(C_devoff, as.integer(which))
##     dev.cur()
## }
## <bytecode: 0x000002907345f828>
## <environment: namespace:grDevices>
```

35. geom_bar

Package:ggplot2

```
#Explanation: Creates a bar chart for two columns
library(ggplot2) #ggplot2 is part of tidyverse


ggplot(data = iris, #data
       aes( #aes stands for aesthetics
         x = Species, #variable for x axis
         fill = Species)) +  #will color bins with subgroups of Species
  geom_bar() #creates a bar chart
```

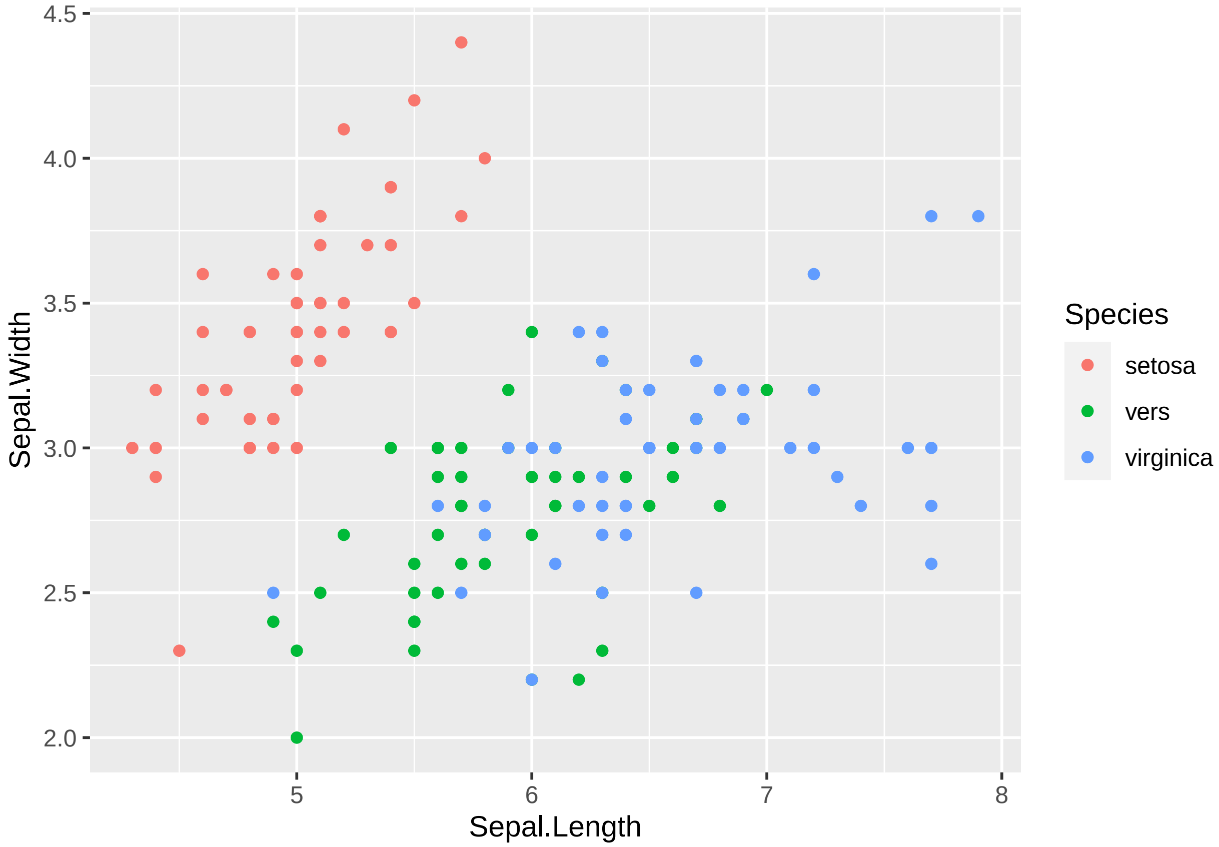


36. geom_point()

Package: ggplot2

```r
#Explanation: Creates a scatter plot for two columns
library(ggplot2) #ggplot2 is part of tidyverse

ggplot(data = iris, #data
       aes(x = Sepal.Length, y = Sepal.Width, #variables for x and y axes
           color = Species)) + #will color bins with subgroups of Species
  geom_point() #creates scatter plot
```



37. scale()

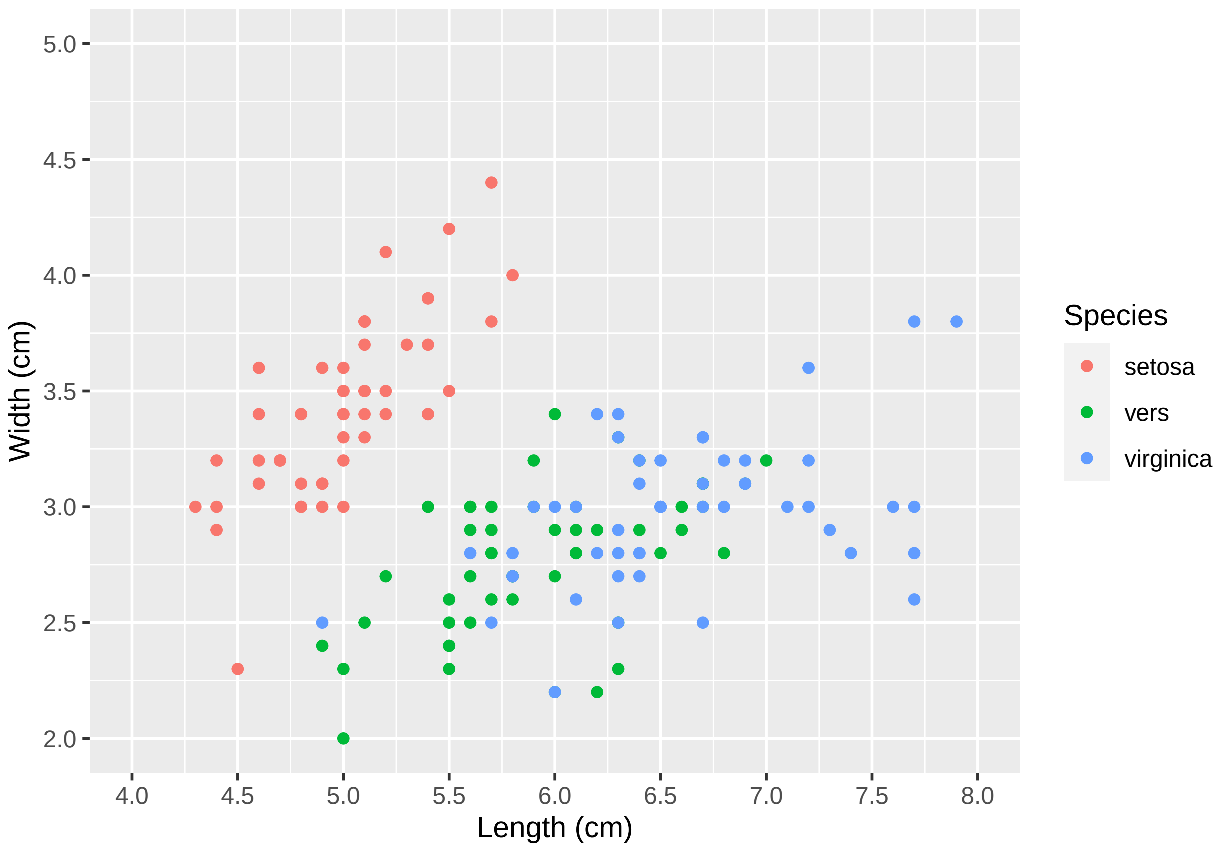
Package: ggplot2

```r
#Explanation: Adjusts scale of axes for scatterplots, bar charts, histrograms and other ggplot visuals
library(ggplot2) #ggplot2 is part of tidyverse

# Create plot
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point() +

# Customize the x-axis scale
scale_x_continuous(name = "Length (cm)", breaks = seq(4, 8, 0.5), limits = c(4, 8)) +

# Customize the y-axis scale
scale_y_continuous(name = "Width (cm)", breaks = seq(2, 5, 0.5), limits = c(2, 5))
```


5.0
4.5
4.0
3.5
3.0
2.5
2.0
Width (cm)
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Length (cm)
Species
setosa
vers
virginica