R Programming Basics - Study Notes

1. Introduction to R

- **R** is a programming language mainly used for **data analysis** and **statistics**.
- It is open-source and has many built-in functions for calculations and data handling.
- RStudio is a popular tool (IDE) used to write and run R programs.

---

2. Assigning Values and Printing

```
x <- 5   # assign 5 to variable x
y = 10  # alternative way to assign
print(x)  # prints value of x
```

- `<-` is the most commonly used assignment operator. - `#` is used to write comments.

---

3. Data Types

- **Numeric**: numbers, e.g. `x <- 10`
- **Character**: text, e.g. `name <- "R Language"`
- **Logical**: TRUE or FALSE values, e.g. `flag <- TRUE`
- **Vector**: a sequence of elements, e.g. `v <- c(1, 2, 3)`
- **Data Frame**: table-like structure, e.g.

  ```
  df <- data.frame(
    name = c("A", "B"),
    age = c(20, 25)
  )
  ```

---

4. Basic Operations

```
2 + 3   # addition
5 - 2   # subtraction
4 * 3   # multiplication
10 / 2  # division

# Sequences
```

```
1:5         # 1, 2, 3, 4, 5
seq(1, 10, by=2)  # 1, 3, 5, 7, 9

# Repetition
rep(5, times=3)  # 5 5 5
```

---

5. Functions

```
my_sum <- function(a, b){
  return(a + b)
}

my_sum(4, 6)  # result = 10
```

- Functions are blocks of code that perform a task. - `function()` is used to create a new function.

---

6. Conditional Statements

```
x <- 10
if(x > 5){
  print("Greater than 5")
} else {
  print("Less or equal to 5")
}
```

- `if` checks a condition and runs code if it is TRUE. - `else` runs code if the condition is FALSE.

---

7. Loops

```
# For loop
for(i in 1:5){
  print(i)
}

# While loop
count <- 1
while(count <= 5){
  print(count)
  count <- count + 1
}
```

- **For loop** repeats code a fixed number of times. - **While loop** repeats code as long as a condition is TRUE.

---

8. Working with Data

```
# View built-in dataset
head(iris)    # first 6 rows
summary(iris)  # summary statistics
```

- `head()` shows the first few rows. - `summary()` gives minimum, maximum, mean, and other details.

---

9. Data Visualization Basics

R provides several simple functions for visualization:

Scatter Plot

```
plot(iris$Sepal.Length, iris$Petal.Length,
     main = "Sepal vs Petal Length",
     xlab = "Sepal Length",
     ylab = "Petal Length",
     col = "blue",
     pch = 19)
```

- Shows the relationship between two numeric variables.

Histogram

```
hist(iris$Sepal.Length,
     main = "Distribution of Sepal Length",
     xlab = "Sepal Length",
     col = "lightgreen",
     border = "black")
```

- Displays the frequency distribution of a variable.

Boxplot

```
boxplot(iris$Sepal.Length ~ iris$Species,
        main = "Sepal Length by Species",
        xlab = "Species",
        ylab = "Sepal Length",
        col = c("orange", "skyblue", "pink"))
```

- Shows spread, median, and outliers across categories.

Line Plot

```
plot(1:10, type="l", col="red", lwd=2,
     main="Line Plot Example",
     xlab="X-axis", ylab="Y-axis")
```

- Useful for showing trends.

Bar Plot

```
counts <- table(iris$Species)
barplot(counts,
        main = "Count of Each Species",
        col = c("purple", "cyan", "yellow"),
        ylab = "Frequency")
```

- Displays counts of categorical data.

10. Conclusion

- R is widely used for **basic calculations, data handling, and visualizations**.
- You can easily make scatter plots, histograms, boxplots, line plots, and bar plots.
- Start by practicing with built-in datasets like **iris** and **mtcars**.
- Learning variables, data types, functions, conditions, loops, data handling, and plotting gives a solid foundation in R.