R Control Flow Statements: Building blocks for automated Decision Making

Hoda Haeri (Matriculation:400963829)

Hochschule Fresenius - University of Applied Science

Author Note

The authors have no conflicts of interest to disclose.

Correspondence concerning this article should be addressed to Hoda Haeri (Matriculation:400963829), Email: haeri.hoda@stud.hs-fresenius.de

Abstract

R is a powerful programming language widely used for data analysis and visualization. Control flow statements in R—such as if, else, for, while, and repeat—allow users to automate decision-making and repetitive tasks. These statements are the core building blocks that enable scripts to respond to data, adapt to changing situations, and streamline complex analytical processes. This document provides a clear overview of the main control flow statements in R, their syntax, and practical examples to illustrate their role in automating decision-making.

Keywords: R programming, Control flow statements, Automation, Data analysis, Decision-making

R Control Flow Statements: Building blocks for automated Decision Making

Table of contents

R Control Flow Statements: Building blocks for automated Decision Making

R is a popular language for statistical analysis, data manipulation, and visualization. One of its key strengths lies in its ability to automate tasks and decision-making processes through the use of control flow statements, such as if, else, for, while, and repeat. These control flow statements act as the logic gates of a program, directing the execution path based on specific conditions or data values. This allows R scripts to handle various types of data, respond to unexpected situations (like missing or unusual values), and efficiently repeat processes without manual intervention. By incorporating control flow statements, programmers can make their code more dynamic, efficient, and capable of adapting to different analytical requirements and datasets . This flexibility is essential for robust data analysis and building scalable data processing pipelines in R.

(Kabacoff, 2022)

What Are Control Flow Statements in R?

Control flow statements in R act much like traffic signals in a city—they control which parts of your code are allowed to execute and when, based on the current state of your data or specific conditions. By using these statements, R programs can make decisions automatically, such as choosing different analysis paths depending on data values. They also allow for repeated actions, like processing every item in a dataset, without requiring manual input each time. Additionally, control flow statements make your code more robust by providing ways to handle unexpected situations, such as missing or unusual data, ensuring that your scripts remain accurate and reliable even when faced with real-world data challenges (Kabacoff, 2022).

Types of Control Flow Statements

1- If, Else If, and Else Statements

- If statements execute a block of code only if a specified condition is true.

Example:

```r
x <- 10


if (x > 5) {
  print("x is greater than 5")
}
```

```
[1] "x is greater than 5"
```

This code will print “x is greater than 5” because the condition is true.

- Else statements execute when the condition in the if statement is false.

Example:

```r
x <- 3


if (x > 5) {
  print("x is greater than 5")
} else {
  print("x is not greater than 5")
}
```

```
[1] "x is not greater than 5"
```

This code will print “x is not greater than 5” because the condition is false and the else block runs.

- Else If statements check additional conditions if the previous conditions were false.

Example:

```r
x <- 5


if (x > 5) {
  print("x is greater than 5")
} else if (x == 5) {
  print("x is exactly 5")
} else {
  print("x is less than 5")
}
```

```
[1] "x is exactly 5"
```

This code will print “x is exactly 5” because the first condition is false but the else if condition is true.

2- Repeat Loops

Repeat loops execute a block of code until a specific condition is met, usually ending with a break statement to prevent infinite looping.

(Kabacoff, 2022)

Example:

```r
count <- 1


repeat {
  print(count)
  count <- count + 1
  if (count > 5) {
    break
  }
}
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

This code prints numbers 1 to 5. The repeat loop keeps running until the condition inside the loop (count > 5) becomes true, which triggers the break statement and stops the loop.

(Grolemund, 2014)

3- While Loops

While loops repeat a block of code as long as a certain condition remains true.

Example:

```
count <- 1


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

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

This code also prints numbers 1 to 5. The loop runs as long as the condition (count <= 5) is true. Once count becomes greater than 5, the loop stops.

(Grolemund, 2014)

4- For Loops

For loops execute a block of code a specific number of times, usually once for each value in a vector.

Example:

```
numbers <- c(1, 2, 3, 4, 5)


for (num in numbers) {
  print(num)
}
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

This code prints each number in the vector numbers, one by one. The loop repeats for every value in the vector.

(R Core Team, 2024)

Why Control Flow Statements Are Important for Automated Decision-Making

Control flow statements make R scripts "think" for themselves, adapting to new data and situations. They allow for automation in data cleaning, handling missing values, and creating decision rules . For example, you can write code that automatically fills in missing values or flags unusual entries—tasks that would otherwise require manual review.

(Kabacoff, 2022)

Conclusion

Control flow statements are fundamental for anyone using R because they turn simple, linear scripts into adaptable and interactive programs. With control flow tools like if, else, for,

while, and repeat, R users can build scripts that make their own decisions and handle repetitive tasks automatically. This capability allows the automation of complex data analysis steps—such as checking for outliers, filling in missing data, or running calculations across entire datasets—without manual intervention. As a result, workflows become not only more efficient but also more powerful, since the code can adjust to changing data and requirements on its own . This adaptability is especially valuable in real-world data analysis, where conditions and data structures often change.

(Kabacoff, 2022) , (Grolemund, 2014)

References

Grolemund, G. (2014). *Hands-on programming with r* (p. 45). O'Reilly Media.

Kabacoff, R. I. (2022). *R in action* (3rd ed., p. 39). Manning Publications.

R Core Team. (2024). *R language definition: Control flow*. R Foundation for Statistical Computing. https://cran.r-project.org/doc/manuals/r-release/R-lang.html#Control-flow

Affidavit

I hereby affirm that this submitted paper was authored unaided and solely by me. Additionally, no other sources than those in the reference list were used. Parts of this paper, including tables and figures, that have been taken either verbatim or analogously from other works have in each case been properly cited with regard to their origin and authorship. This paper either in parts or in its entirety, be it in the same or similar form, has not been submitted to any other examination board and has not been published.

I acknowledge that the university may use plagiarism detection software to check my thesis. I agree to cooperate with any investigation of suspected plagiarism and to provide any additional information or evidence requested by the university.

Checklist:

- ☒ The handout contains 3-5 pages of text.
- ☒ The submission contains the Quarto file of the handout.

- ☒ The submission contains the Quarto file of the presentation.
- ☒ The submission contains the HTML file of the handout.
- ☒ The submission contains the HTML file of the presentation.
- ☒ The submission contains the PDF file of the handout.
- ☒ The submission contains the PDF file of the presentation.
- ☒ The title page of the presentation and the handout contain personal details (name, email, matriculation number).
- ☒ The handout contains a abstract.
- ☒ The presentation and the handout contain a bibliography, created using BibTeX with APA citation style.
- ☒ Either the handout or the presentation contains R code that proof the expertise in coding.
- ☒ The handout includes an introduction to guide the reader and a conclusion summarizing the work and discussing potential further investigations and readings, respectively.
- ☒ All significant resources used in the report and R code development.
- ☒ The filled out Affidavit.
- ☒ A concise description of the successful use of Git and GitHub, as detailed here: https://github.com/hubchev/make_a_pull_request.
- ☒ The link to the presentation and the handout published on GitHub.

[Hoda Haeri,] [06/18/2025,] [Koln]