R Programming Data Structures, Control Flow, Functions, and Visualization An Example-Driven Introduction

Mehmet Tarık ATAY
Mehmet Barış DURMUŞ
Fatih GÜLBAHAR

PEGEM AKADEMİ

Mehmet Tarık ATAY • Mehmet Barış DURMUŞ • Fatih GÜLBAHAR

R PROGRAMMING DATA STRUCTURES, CONTROL FLOW, FUNCTIONS, AND VISUALIZATION AN EXAMPLE-DRIVEN INTRODUCTION

ISBN 978-625-8897-65-4

Responsibility of the contents belongs to its authors.

© 2026, PEGEM AKADEMİ

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage or retrieval system, without prior permission in writing from the publishers.

Pegem Academy Publishing Training and Consultancy Services Tic. Ltd. Şti.It is a publishing house founded in 1998 in Ankara/Turkey which has been operating regularly for 27 years. Published books, it is included in the catalogs of higher education institutions. Pegem Academy has over 2000 publications in the same field from different authors. Detailed information about our publications can be found http://pegem.net.

1st Edition: November 2026, Ankara

Publication-Project: Selcan Özgürel
Typesetter-Graphic Designer: Beyza Nur Erdoğan
Cover Design: Pegem Akademi

Printed by: Sayfa Basım Sanayi Ticaret Ltd. Şti.
İvedik Organized Industrial Zone, Printing Houses Site,
1514th Street Number, No: 23-25,
Yenimahalle/ANKARA

Publishing House Certificate Number: 51818
Printing House Certificate Number: 77079

Contact

Pegem Akademi: Shira Trade Center
Macun Neighbourhood 204th Street Number: 141/A-33 Yenimahalle/ANKARA
Publishig House: 0312 430 67 50
Distribution: 0312 434 54 24
Preparatory Courses: 0312 419 05 60
Internet: www.pegem.net
E-mail: yayinevi@pegem.net
WhatsApp Line: 0538 594 92 40

Preface

Over the past two decades—and particularly during the last five years—computer technologies and their applications have expanded rapidly and significantly into almost every aspect of our daily and scientific lives. As a result, the range of software tools used in statistics and data science has become increasingly diverse and powerful. Examples include Excel, SPSS, Weka, and Python. In recent years, the R programming language and its associated tools have also become widely used by researchers and practitioners in statistics and data science.

From this perspective, the R programming language was originally developed primarily for statistical computing and scientific research and has since been widely adopted across numerous disciplines. R is an open-source, freely available programming language supported and continuously developed by a large international community. Its extensibility, together with the fact that it is a programming language rather than merely a statistical software package, makes it highly flexible and adaptable to a wide variety of research and analytical purposes.

R is widely used in many fields, including data mining, artificial intelligence, artificial neural networks, web mining, deep learning, and statistical analysis.

This book is designed for readers who are beginning to learn R, a popular environment for statistical computing, together with the RStudio integrated development environment. The material is presented at a level and with sufficient detail to help beginners understand the concepts and apply them with ease. Although numerous written and visual learning resources on R programming are available, relatively few systematically guide students from the fundamentals of the language toward more advanced applications.

Another important consideration in preparing this book was to make the content easy for students to follow. For this reason, the book adopts an example-oriented, application-based introductory structure. Fundamental topics such as Data Structures, Control Flow, Functions, and Visualization are introduced and explained primarily through practical examples.

In this respect, the applications included in the book have been selected according to a "Cookbook" approach to learning R programming. Rather than focusing solely on theoretical explanations, the book demonstrates how concepts can be understood and applied through carefully selected examples and practical exercises.

As researchers and academics, we have placed particular emphasis on developing this book from a learning-by-doing, practice-oriented perspective. Our primary objective is to provide readers with a resource that is clear and easy to follow and that enables them to progressively develop their knowledge, practical skills, and competencies in R programming.

Assoc. Prof. Dr. Mehmet Tarık Atay (**METU – Mathematics, B.Sc.**)

Mehmet Barış Durmuş (**METU – Statistics, B.Sc.**)

Fatih Gülbahar (**METU – Statistics, B.Sc.**)

Acknowledgements

To my father, Atty. İsmail İhsan Atay, and my mother, Hatice Atay, a devoted teacher:

Thank you for continuing to be a light in my life, for guiding me along my path, and, above all, for laying the foundations that have inspired me to strive to become both a good person and a dedicated scholar.

With my deepest gratitude,

Assoc. Prof. Dr. Mehmet Tarık Atay

Dedicated to:

My mother, Cennet Durmuş, and my father, Hacı Emin Durmuş, who devoted their lives to shaping who I am today with boundless love and patience; Nisa Durmuş—the princess of our family—who has been both a sister and a friend; Ayşenur Yıldırım, my precious supporter and companion, for standing by me through every step of this journey; my esteemed professor Mehmet Tarık Atay, whose invaluable guidance and mentorship illuminated this work; my dear friend Fatih Gülbahar, for his partnership and dedication in making this book a reality; and everyone striving to shape the future.

Mehmet Barış Durmuş

My heartfelt thanks go to those who supported me throughout this journey:

İbrahim Tümay Gülbahar, not just a brother but also a mentor who helped me at every step; my mother Derya Gülbahar and my father Üzeyir Gülbahar for their unconditional love and support; my esteemed professor Mehmet Tarık Atay, whose invaluable guidance brought me to where I am today; Mehmet Barış Durmuş, for our long-time friendship and the hard work he put into this book; my professors at METU, who prepared me for the challenges of academic and professional life; and my colleagues at Türkiye İş Bankası—Team Bumblebee—whose support made continuing my academic journey possible.

I hope this book serves those who are beginning their own learning journeys; knowing it helped will make all our efforts worthwhile.

Fatih Gülbahar

1 Vectors and Factors: Fundamental Building Blocks 1
1.1 Vector Indexing 2
1.2 Vector Calculations 2
1.3 Vector Addition 3
1.4 Logical Operators and Vector Comparisons 4
1.5 Character Vectors 5
1.6 Vector Length and Character Counts 5
1.7 Combining Vectors 6
1.8 Naming Vector Elements 7
1.9 Conditional Queries on Vectors 7
1.10 Adding, Deleting, and Changing Vector Elements 8
1.11 Summary Operations 10
1.12 Sorting and Ordering 11
1.13 Factors 12
1.14 Factors from Numeric Vectors 12
1.15 Modifying Factor Values 13
1.16 Ordered Factors 13
1.17 Removing Observations from Factors 13
1.18 Checking Data Types 14
2 Arrays and Matrices 17
2.1 Arrays 17
2.2 Creating Arrays with array() 18
2.3 Array Indexing and Slicing 19
2.4 Batch Operations on Arrays with apply() 19
2.5 Matrices 20
2.6 Creating Matrices 20
2.7 Naming Matrix Dimensions 20
2.8 Inspecting Matrix Dimensions 21
2.9 Identity Matrices 22
2.10 Matrix Indexing and Slicing 22
2.11 Matrix Operations 23
2.12 Matrix Multiplication 24
2.13 Matrix Transpose 24
2.14 Determinants 25

2.15 Matrix Inverse 25
2.16 Logical Selection in Matrices 26
2.17 Changing Matrix Elements 27
2.18 Deleting Rows and Columns 28
2.19 Adding Rows and Columns 28
2.20 Merging Matrices 29
2.21 Checking Type and Element Count 30
2.22 Minimum and Maximum Values 30
2.23 Row and Column Sums 31
2.24 Row and Column Means 32
2.25 Applying Functions over Matrix Margins 32
2.26 Diagonal Elements 33
3 Lists: Heterogeneous Data Structures 34
3.1 Creating a List 34
3.2 Further List Creation 34
3.3 Lists Containing Vectors, Data Frames, and Matrices 35
3.4 Creating a List from a Vector 35
3.5 Creating an Empty List 36
3.6 Accessing List Elements 36
3.7 Slicing a List 38
3.8 Selecting Multiple Components 39
3.9 Extracting Objects from a List 39
3.10 Using List Elements Directly 40
3.11 Safe Evaluation with with() 41
3.12 Adding Elements to a List 41
3.13 Inserting Elements with append() 43
3.14 Deleting List Elements 44
3.15 Changing Elements Inside a List 46
3.16 Applying Functions to List Elements 47
3.17 Using lapply() 48
3.18 Using sapply() 48
3.19 lapply() versus sapply() 48
3.20 Flattening Lists with unlist() 49
3.21 Naming List Components 49

3.22 Nested Lists 50
3.23 Lists from Text Processing 50
4 Character Vectors and String Operations 52
4.1 Creating Character Objects and Vectors 52
4.2 Empty Character Vectors 52
4.3 Measuring Character Length 53
4.4 Type Checks and Class Conversion 54
4.5 Combining Character Objects with paste() 56
4.6 Changing Letter Case 57
4.7 Abbreviating Character Values 58
4.8 Character Translation with chartr() 60
4.9 Working with Substrings 60
4.10 Printing and Concatenating Character Output 61
4.11 Formatting Numeric and Character Output 63
4.12 Converting Values with toString() 65
4.13 Searching Character Data and Patterns 66
4.14 Replacing and Extracting Patterns 69
4.15 Trimming and Normalizing Whitespace 70
4.16 Case Study: Data Cleaning with Regex and stringr 71
5 Data Frames and Applied Functions 72
5.1 Creating a Data Frame 72
5.2 Indexing and Subsetting 72
5.3 Summarizing a Data Frame 73
5.4 Conditional Subsetting 73
5.5 Viewing Data in RStudio 73
5.6 Accessing Variables with the Dollar Operator and Checking Dimensions 73
5.7 Creating an Empty Data Frame 74
5.8 Naming Rows and Columns 74
5.9 Renaming a Selected Column 75
5.10 Reviewing Data Frames 75
5.11 Inspecting Structure with str() 75
5.12 Inspecting Attributes 75
5.13 Accessing Specific Components 75
5.14 Applying Functions to Data Frames 76

5.15 Conversions Between Matrices, Lists, and Data Frames 76
6 Graphics and Data Visualization 78
6.1 Graph Selection Guideline 78
6.2 Axis Customization and Graphical Parameters 78
6.2.1 Point Shapes, Line Styles, and Annotations 78
6.2.2 Graphic Magnification and Background Customization 79
6.3 Scatter Plots and Trend Lines 80
6.3.1 Group-Disaggregated Scatter Plot 81
6.4 Bubble Plots: Multi-Dimensional Data 82
6.5 Bar Charts, Mosaic Plots, and Pie Charts 84
6.6 Histograms and Density Distribution Analysis 88
6.7 Boxplots and Distribution Comparisons 90
6.8 Advanced Visualization with ggplot2 92
7 Control Structures and Loops 95
7.1 Comparison Operators 95
7.2 Logical Operators 95
7.3 if Statements 96
7.4 if-else and else-if Structures 97
7.5 Nested Conditional Selection 97
7.6 Vectorized ifelse() 98
7.7 for Loops 98
7.8 Vectorization versus Loops 100
7.9 while Loops 100
7.10 Factorials with Loops 101
7.11 Ending a Loop with break 101
7.12 The Same Procedure with while 102
7.13 Continuing with next 102
7.14 Complex Logical Conditions in for Loops 103
7.15 next with while Loops 103
7.16 repeat Loops 104
7.17 Nested Loops 104
7.18 Combining Loops with Vectors 105
7.19 Using which() as an Alternative 105
7.20 Error Handling with tryCatch() 106

7.21 Vector Addition without Loops . . . 107
7.22 Writing a Simple Function . . . 107
7.23 apply() with Data Frames . . . 107
7.24 apply() with Matrices . . . 108
7.25 summary() by Columns . . . 109
7.26 summary() by Rows . . . 109
7.27 Exam-Score Matrix Example . . . 110
7.28 lapply() . . . 112
7.29 sapply() and vapply() . . . 113
7.30 Combining Experimental Results in Lists . . . 114
7.31 Transposing Results with lapply() . . . 115
7.32 A Data-Frame Example . . . 116
7.33 Computing Column Means with apply() . . . 116
7.34 Categorical Operations with tapply() . . . 116
7.35 Selection with switch() . . . 117
7.36 Performance Tip: Memory Pre-allocation . . . 117
8 Functions and User-Defined Functions . . . 119
8.1 Defining a Basic Function . . . 119
8.2 Functions for Mathematical Calculations . . . 119
8.3 Nested Functions . . . 120
8.4 Flexible Arguments with the Ellipsis . . . 120
8.5 Return Values . . . 121
8.6 Implicit Return Values . . . 121
8.7 Explicit return() . . . 122
8.8 Returning Lists . . . 122
8.9 Default Arguments . . . 122
8.10 Functions with Conditional Logic . . . 123
8.11 Recursive Functions . . . 124
8.12 A Fibonacci Sequence Generator . . . 125
8.13 Alternative Fibonacci Code . . . 125
8.14 Adding Arguments to Functions . . . 126
8.15 Returning a Generated Sequence . . . 127
8.16 Function Termination after return() . . . 129
8.17 A Shorter Factorial Function . . . 129

8.18 The Built-In factorial() Function 130
8.19 Converting Numeric Grades to Letter Grades 130
8.20 Built-In Mathematical Functions 131
8.21 Trigonometric Example 131
8.22 Vectorization: The R Way 131
8.23 Using log10() 132
8.24 Using exp() 132
9 Data Input and Output in R 134
9.1 Loading Data-Reading Packages 134
9.2 Entering Data with Data Frames and Lists 134
9.3 Entering Data with the R Command Editor 135
9.4 Entering Numeric Values with scan() 135
9.5 Entering Character and Numeric Values with scan() 136
9.6 Choosing a Text-File Reader 137
9.7 Reading Fixed-Width Text Files 137
9.8 Using R Commander for Data Entry 137
9.9 Entering Data from the Keyboard or Terminal 138
9.10 Working Directories and File Locations 139
9.11 Listing Files in the Current Directory 139
9.12 Reading Delimited Files 139
9.13 Reading Semicolon-Separated CSV Files 140
9.14 Reading Excel Files 141
9.15 Reading Tables 141
9.16 Writing Data with R 141
9.17 Writing Tables with write.table() 141
9.18 Writing CSV Files 142
9.19 Writing with cat() 143
9.20 Writing Lines with writeLines() 143
9.21 Capturing Output with sink() 143
9.22 Saving Native R Objects 144
9.23 High-Performance I/O with data.table 144
9.24 Missing Values, Relative Paths, and Encoding 145
9.25 Reading JSON Data from Web APIs 145
9.26 Connecting to SQL Databases 146

Chapter 1

Vectors and Factors: Fundamental Building Blocks

Vectors are the fundamental one-dimensional data structure in R. This chapter introduces vector creation, indexing, arithmetic, logical comparisons, character data, modification, summary functions, sorting, and factors through practical examples.

Vectors can be considered as one dimensional sequences to create vectors use c() command.

```
v <- c(4,6,-5,-1,7,0,6,10,-15)
```

```
v
```

```
[1]   4   6  -5  -1   7   0   6  10 -15
```

We can get the same result by using assign() command also.

```
assign("v2", c(2,45,-3,14-9,3,0,11))
```

```
v2
```

```
[1]  2 45 -3  5  3  0 11
```

We can create another vector by using previously assigned vector as a part of the newly to be created vector.

```
w <- c(11,21,-5, v2)
```

```
w
```

```
[1] 11 21 -5  2 45 -3  5  3  0 11
```

All the members of a vector should be in the same type as numeric or character or logical types. etc.

```
a <- c(30, 22.5, FALSE, TRUE, " Mehmet", "JONY")
```

```
a
```