



# Functional Programming in R

Advanced Statistical Programming for  
Data Science, Analysis and Finance

—  
Thomas Mailund

Apress®

# Functional Programming in R

Advanced Statistical  
Programming for Data Science,  
Analysis and Finance



**Thomas Mailund**

Apress®

***Functional Programming in R: Advanced Statistical Programming for Data Science, Analysis and Finance***

Thomas Mailund  
Aarhus N, Denmark

ISBN-13 (pbk): 978-1-4842-2745-9  
DOI 10.1007/978-1-4842-2746-6

ISBN-13 (electronic): 978-1-4842-2746-6

Library of Congress Control Number: 2017937314

Copyright © 2017 by Thomas Mailund

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr  
Editorial Director: Todd Green  
Acquisitions Editor: Steve Anglin  
Development Editor: Matthew Moodie  
Technical Reviewer: Andrew Moskowitz  
Coordinating Editor: Mark Powers  
Copy Editor: Mary Bearden  
Compositor: SPi Global  
Indexer: SPi Global  
Artist: SPi Global  
Cover Image designed by Freepik

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com/rights-permissions](http://www.apress.com/rights-permissions).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at <http://www.apress.com/bulk-sales>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/9781484227459](http://www.apress.com/9781484227459). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

# Contents at a Glance

<b>About the Author .....</b>	<b>ix</b>
<b>About the Technical Reviewer .....</b>	<b>xi</b>
<b>Acknowledgments .....</b>	<b>xiii</b>
<b>Introduction .....</b>	<b>xv</b>
<b>■ Chapter 1: Functions in R .....</b>	<b>1</b>
<b>■ Chapter 2: Pure Functional Programming .....</b>	<b>25</b>
<b>■ Chapter 3: Scope and Closures.....</b>	<b>43</b>
<b>■ Chapter 4: Higher-Order Functions.....</b>	<b>63</b>
<b>■ Chapter 5: Filter, Map, and Reduce.....</b>	<b>77</b>
<b>■ Chapter 6: Point-Free Programming.....</b>	<b>95</b>
<b>Afterword.....</b>	<b>101</b>
<b>Index.....</b>	<b>103</b>

# Contents

<b>About the Author .....</b>	<b>ix</b>
<b>About the Technical Reviewer .....</b>	<b>xi</b>
<b>Acknowledgments .....</b>	<b>xiii</b>
<b>Introduction .....</b>	<b>xv</b>
<b>■ Chapter 1: Functions in R .....</b>	<b>1</b>
Writing Functions in R .....	1
Named Parameters and Default Parameters .....	3
The “Gobble Up Everything Else” Parameter: .....	4
Functions Don’t Have Names.....	6
Lazy Evaluation .....	7
Vectorized Functions .....	12
Infix Operators .....	16
Replacement Functions.....	18
<b>■ Chapter 2: Pure Functional Programming .....</b>	<b>25</b>
Writing Pure Functions .....	26
Recursion as Loops .....	27
The Structure of a Recursive Function .....	30
Tail-Recursion .....	37
Runtime Considerations .....	38

■ <b>Chapter 3: Scope and Closures</b> .....	<b>43</b>
Scopes and Environments.....	43
Environment Chains, Scope, and Function Calls .....	46
Scopes, Lazy Evaluation, and Default Parameters .....	51
Nested Functions and Scopes.....	53
Closures .....	56
Reaching Outside Your Innermost Scope .....	57
Lexical Scope and Dynamic Scope.....	59
■ <b>Chapter 4: Higher-Order Functions</b> .....	<b>63</b>
Currying.....	65
A Parameter Binding Function.....	69
Continuation-Passing Style .....	70
Thunks and Trampolines .....	72
■ <b>Chapter 5: Filter, Map, and Reduce</b> .....	<b>77</b>
The General Sequence Object in R Is a List.....	77
Filtering Sequences.....	79
Mapping Over Sequences .....	80
Reducing Sequences.....	82
Bringing the Functions Together .....	84
The Apply Family of Functions .....	87
sapply, vapply, and lapply .....	87
The apply Function .....	88
The tapply Function .....	89

Functional Programming in purrr .....	90
Using library(purrr).....	90
Filter-like Functions.....	90
Map-like Functions.....	91
Reduce-like Functions.....	93
<b>■ Chapter 6: Point-Free Programming.....</b>	<b>95</b>
Function Composition.....	95
Pipelines.....	97
<b>Afterword.....</b>	<b>101</b>
<b>Index.....</b>	<b>103</b>

# About the Author

**Thomas Mailund** is an associate professor in bioinformatics at Aarhus University, Denmark. His background is in math and computer science, but for the past decade his main focus has been on genetics and evolutionary studies, particularly comparative genomics, speciation, and gene flow between emerging species.



# About the Technical Reviewer



**Andrew Moskowitz** is a doctoral candidate in quantitative psychology at UCLA and a self-employed statistical consultant. His quantitative research focuses mainly on hypothesis testing and effect sizes in mixed effects models. While at UCLA, Andrew has collaborated with a number of faculty, students, and enterprises to help them derive meaning from data across an array of fields, ranging from psychological services and health care delivery to marketing.

# Acknowledgments

I would like to thank Duncan Murdoch and the people on the R-help mailing list for helping me work out a kink in lazy evaluation in the trampoline example.

# Introduction

Welcome to *Functional Programming in R*! I wrote this book, to have teaching material beyond the typical introductory level most textbooks on R have. This book is intended to give an introduction to functions in R and how to write functional programs in R. Functional programming is a style of programming, like object-oriented programming, but one that focuses on data transformations and calculations rather than objects and state.

Where in object-oriented programming you model your programs by describing which states an object can be in and how methods will reveal or modify that state, in functional programming you model programs by describing how functions translate input data to output data. Functions themselves are considered data that you can manipulate, and much of the strength of functional programming comes from manipulating functions, building more complex functions by combining simpler functions.

The R programming language supports both object-oriented programming and functional programming, but it is mainly a functional language. It is not a “pure” functional language. Pure functional languages will not allow you to modify the state of the program by changing the values parameters hold and will not allow functions to have side effects (and need various tricks to deal with program input and output because of it).

R is somewhat closest to “pure” functional languages. In general, data are immutable, so changes to data inside a function do not ordinarily alter the state of data outside that function. But R does allow side effects, such as printing data or making plots, and of course it allows variables to change values.

Pure functions are functions that have no side effects and where a function called with the same input will always return the same output. Pure functions are easier to debug and to reason with because of this. They can be reasoned with in isolation and will not depend on the context in which they are called. The R language does not guarantee that the functions you write are pure, but you can write most of your programs using only pure functions. By keeping your code mostly purely functional, you will write more robust code and code that is easier to modify when the need arises.

You will just have to move the impure functions to a small subset of your program. These functions are typically those that need to sample random data or that produce output (either text or plots). If you know where your impure functions are, you know when to be extra careful with modifying code.

Chapter 1 contains a short introduction to functions in R. Some parts you might already know, so in that case feel free to skip ahead, but I give an exhaustive description of how functions are defined and used to make sure that we are all on the same page. The following chapters then move on to more complex issues.