



Quick answers to common problems

Haskell Data Analysis Cookbook

Explore intuitive data analysis techniques and powerful machine learning methods using over 130 practical recipes

Nishant Shukla

[PACKT] open source 
PUBLISHING community excellence defined

Table of Contents

[Haskell Data Analysis Cookbook](#)

[Credits](#)

[About the Author](#)

[About the Reviewers](#)

[www.PacktPub.com](#)

[Support files, eBooks, discount offers, and more](#)

[Why Subscribe?](#)

[Free Access for Packt account holders](#)

[Preface](#)

[What this book covers](#)

[What you need for this book](#)

[Who this book is for](#)

[Conventions](#)

[Reader feedback](#)

[Customer support](#)

[Downloading the example code](#)

[Errata](#)

[Piracy](#)

[Questions](#)

[1. The Hunt for Data](#)

[Introduction](#)

[Harnessing data from various sources](#)

[How to do it...](#)

[News](#)

[Private](#)

[Academic](#)

[Nonprofits](#)

[The United States government](#)

[Accumulating text data from a file path](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Catching I/O code faults](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Keeping and representing data from a CSV file](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Examining a JSON file with the aeson package](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Reading an XML file using the HXT package](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Capturing table rows from an HTML page](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Understanding how to perform HTTP GET requests](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also...](#)

[Learning how to perform HTTP POST requests](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Traversing online directories for data](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Using MongoDB queries in Haskell](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Reading from a remote MongoDB server](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Exploring data from a SQLite database](#)

[Getting ready](#)

[How to do it...](#)

[2. Integrity and Inspection](#)

[Introduction](#)

[Trimming excess whitespace](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Ignoring punctuation and specific characters](#)

[How to do it...](#)

[There's more...](#)

[Coping with unexpected or missing input](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Validating records by matching regular expressions](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Lexing and parsing an e-mail address](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Deduplication of nonconflicting data items](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Deduplication of conflicting data items](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Implementing a frequency table using Data.List](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Implementing a frequency table using Data.MultiSet](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Computing the Manhattan distance](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Computing the Euclidean distance](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Comparing scaled data using the Pearson correlation coefficient](#)

[How to do it...](#)

[How it works...](#)

[Comparing sparse data using cosine similarity](#)

[How to do it...](#)

[See also](#)

[3. The Science of Words](#)

[Introduction](#)

[Displaying a number in another base](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Reading a number from another base](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Searching for a substring using Data.ByteString](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Searching a string using the Boyer-Moore-Horspool algorithm](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Searching a string using the Rabin-Karp algorithm](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Splitting a string on lines, words, or arbitrary tokens](#)

[Getting ready](#)

[How to do it...](#)

[Finding the longest common subsequence](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Computing a phonetic code](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Computing the edit distance](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Computing the Jaro-Winkler distance between two strings](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Finding strings within one-edit distance](#)

[Getting ready](#)

[How to do it...](#)

[There's more...](#)

[See also](#)

[Fixing spelling mistakes](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[4. Data Hashing](#)

[Introduction](#)

[Hashing a primitive data type](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Hashing a custom data type](#)

[Getting ready](#)

[How to do it...](#)

[There's more...](#)

[See also](#)

[Running popular cryptographic hash functions](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Running a cryptographic checksum on a file](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Performing fast comparisons between data types](#)

[How to do it...](#)

[Using a high-performance hash table](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Using Google's CityHash hash functions for strings](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Computing a Geohash for location coordinates](#)

[Getting ready](#)

[How to do it...](#)

[Using a bloom filter to remove unique items](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Running MurmurHash, a simple but speedy hashing algorithm](#)

[Getting ready](#)

[How to do it...](#)

[Measuring image similarity with perceptual hashes](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[5. The Dance with Trees](#)

[Introduction](#)

[Defining a binary tree data type](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Defining a rose tree \(multiway tree\) data type](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Traversing a tree depth-first](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Traversing a tree breadth-first](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Implementing a Foldable instance for a tree](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Calculating the height of a tree](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Implementing a binary search tree data structure](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Verifying the order property of a binary search tree](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Using a self-balancing tree](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Implementing a min-heap data structure](#)

[Getting started](#)

[How to do it...](#)

[There's more...](#)

[Encoding a string using a Huffman tree](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Decoding a Huffman code](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[6. Graph Fundamentals](#)

[Introduction](#)

[Representing a graph from a list of edges](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Representing a graph from an adjacency list](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Conducting a topological sort on a graph](#)

[Getting ready](#)

[How to do it...](#)

[Traversing a graph depth-first](#)

[How to do it...](#)

[Traversing a graph breadth-first](#)

[How to do it...](#)

[Visualizing a graph using Graphviz](#)

[Getting ready](#)

[How to do it...](#)

[Using Directed Acyclic Word Graphs](#)

[Getting ready](#)

[How to do it...](#)

[Working with hexagonal and square grid networks](#)

[Getting started](#)

[How to do it...](#)

[Finding maximal cliques in a graph](#)

[Getting started](#)

[How to do it...](#)

[How it works...](#)

[Determining whether any two graphs are isomorphic](#)

[Getting started](#)

[How to do it...](#)

[7. Statistics and Analysis](#)

[Introduction](#)

[Calculating a moving average](#)

[Getting ready](#)

[How to do it...](#)

[There's more...](#)

[See also](#)

[Calculating a moving median](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Approximating a linear regression](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Approximating a quadratic regression](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Obtaining the covariance matrix from samples](#)

[Getting ready](#)

[How to do it...](#)

[Finding all unique pairings in a list](#)

[How it works...](#)

[See also](#)

[Using the Pearson correlation coefficient](#)

[Getting ready](#)

[How to do it...](#)

[Evaluating a Bayesian network](#)

[Getting ready](#)

[How to do it...](#)

[Creating a data structure for playing cards](#)

[Getting ready](#)

[How to do it...](#)

[Using a Markov chain to generate text](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Creating n-grams from a list](#)

[How to do it...](#)

[Creating a neural network perceptron](#)

[Getting ready](#)

[How to do it...](#)

[8. Clustering and Classification](#)

[Introduction](#)

[Implementing the k-means clustering algorithm](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Implementing hierarchical clustering](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using a hierarchical clustering library](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Finding the number of clusters](#)

[Getting ready](#)

[How to do it...](#)

[Clustering words by their lexemes](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Classifying the parts of speech of words](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Identifying key words in a corpus of text](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Training a parts-of-speech tagger](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Implementing a decision tree classifier](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Implementing a k-Nearest Neighbors classifier](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Visualizing points using Graphics.EasyPlot](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[9. Parallel and Concurrent Design](#)

[Introduction](#)

[Using the Haskell Runtime System options](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Evaluating a procedure in parallel](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Controlling parallel algorithms in sequence](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Forking I/O actions for concurrency](#)

[How to do it...](#)

[See also](#)

[Communicating with a forked I/O action](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Killing forked threads](#)

[How to do it...](#)

[How it works...](#)

[Parallelizing pure functions using the Par monad](#)

[Getting ready](#)

[How to do it...](#)

[There's more...](#)

[See also](#)

[Mapping over a list in parallel](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Accessing tuple elements in parallel](#)

[How to do it...](#)

[There's more...](#)

[See also](#)

[Implementing MapReduce to count word frequencies](#)

[Getting ready](#)

[How to do it...](#)

[Manipulating images in parallel using Repa](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Benchmarking runtime performance in Haskell](#)

[How to do it...](#)

[See also](#)

[Using the criterion package to measure performance](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[Benchmarking runtime performance in the terminal](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

10. Real-time Data

Introduction

Streaming Twitter for real-time sentiment analysis

Getting ready

How to do it...

How it works...

There's more...

Reading IRC chat room messages

Getting ready

How to do it...

See also

Responding to IRC messages

Getting ready

How to do it...

See also

Polling a web server for latest updates

How to do it...

Detecting real-time file directory changes

Getting ready

How to do it...

How it works...

Communicating in real time through sockets

How to do it...

How it works...

Detecting faces and eyes through a camera stream

Getting ready

How to do it...

How it works...

Streaming camera frames for template matching

Getting ready

How to do it...

There's more...

11. Visualizing Data

Introduction

Plotting a line chart using Google's Chart API

Getting ready

How to do it...

How it works...

There's more...

See also

Plotting a pie chart using Google's Chart API

Getting ready

How to do it...

How it works...

There's more...

[See also](#)

[Plotting bar graphs using Google's Chart API](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Displaying a line graph using gnuplot](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Displaying a scatter plot of two-dimensional points](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Interacting with points in a three-dimensional space](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Visualizing a graph network](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Customizing the looks of a graph network diagram](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Rendering a bar graph in JavaScript using D3.js](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Rendering a scatter plot in JavaScript using D3.js](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Diagramming a path from a list of vectors](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[12. Exporting and Presenting](#)

[Introduction](#)

[Exporting data to a CSV file](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Exporting data as JSON](#)

[Getting ready](#)

[How to do it...](#)

[There's more...](#)

[See also](#)

[Using SQLite to store data](#)

[Getting Ready](#)

[How to do it...](#)

[See also](#)

[Saving data to a MongoDB database](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Presenting results in an HTML web page](#)

[Getting ready](#)

[How to do it...](#)

[See also](#)

[Creating a LaTeX table to display results](#)

[Getting Ready](#)

[How to do it...](#)

[See also](#)

[Personalizing messages using a text template](#)

[Getting ready](#)

[How to do it...](#)

[Exporting matrix values to a file](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Index](#)

Haskell Data Analysis Cookbook

Haskell Data Analysis Cookbook

Copyright © 2014 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: June 2014

Production reference: 1180614

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78328-633-1

www.packtpub.com

Cover image by Jarek Blaminsky (<milak6@wp.pl>)

Credits

Author

Nishant Shukla

Reviewers

Lorenzo Bolla

James Church

Andreas Hammar

Marisa Reddy

Commissioning Editor

Akram Hussain

Acquisition Editor

Sam Wood

Content Development Editor

Shaon Basu

Technical Editors

Shruti Rawool

Nachiket Vartak

Copy Editors

Sarang Chari

Janbal Dharmaraj

Gladson Monteiro

Deepa Nambiar

Karuna Narayanan

Alfida Paiva

Project Coordinator

Mary Alex

Proofreaders

Paul Hindle

Jonathan Todd

Bernadette Watkins

Indexer

Hemangini Bari

Graphics

Sheetal Aute

Ronak Dhruv

Valentina Dsilva

Disha Haria

Production Coordinator

Arvindkumar Gupta

Cover Work

Arvindkumar Gupta

About the Author

Nishant Shukla is a computer scientist with a passion for mathematics. Throughout the years, he has worked for a handful of start-ups and large corporations including WillowTree Apps, Microsoft, Facebook, and Foursquare.

Stepping into the world of Haskell was his excuse for better understanding Category Theory first, but eventually, he found himself immersed in the language. His semester-long introductory Haskell course in the engineering school at the University of Virginia (<http://shuklan.com/haskell>) has been accessed by individuals from over 154 countries around the world, gathering over 45,000 unique visitors.

Besides Haskell, he is a proponent of decentralized Internet and open source software. His academic research in the fields of Machine Learning, Neural Networks, and Computer Vision aim to supply a fundamental contribution to the world of computing.

Between discussing primes, paradoxes, and palindromes, it is my delight to invent the future with Marisa.

With appreciation beyond expression, but an expression nonetheless—thank you Mom (Suman), Dad (Umesh), and Natasha.

About the Reviewers

Lorenzo Bolla holds a PhD in Numerical Methods and works as a software engineer in London. His interests span from functional languages to high-performance computing to web applications. When he's not coding, he is either playing piano or basketball.

James Church completed his PhD in Engineering Science with a focus on computational geometry at the University of Mississippi in 2014 under the advice of Dr. Yixin Chen. While a graduate student at the University of Mississippi, he taught a number of courses for the Computer and Information Science's undergraduates, including a popular class on data analysis techniques. Following his graduation, he joined the faculty of the University of West Georgia's Department of Computer Science as an assistant professor. He is also a reviewer of *The Manga Guide To Regression Analysis*, written by *Shin Takahashi, Iroha Inoue, and Trend-Pro Co. Ltd.*, and published by *No Starch Press*.

I would like to thank Dr. Conrad Cunningham for recommending me to Packt Publishing as a reviewer.

Andreas Hammar is a Computer Science student at Norwegian University of Science and Technology and a Haskell enthusiast. He started programming when he was 12, and over the years, he has programmed in many different languages. Around five years ago, he discovered functional programming, and since 2011, he has contributed over 700 answers in the Haskell tag on Stack Overflow, making him one of the top Haskell contributors on the site. He is currently working part time as a web developer at the Student Society in Trondheim, Norway.

Marisa Reddy is pursuing her B.A. in Computer Science and Economics at the University of Virginia. Her primary interests lie in computer vision and financial modeling, two areas in which functional programming is rife with possibilities.

I congratulate Nishant Shukla for the tremendous job he did in writing this superb book of recipes and thank him for the opportunity to be a part of the process.

Support files, eBooks, discount offers, and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

The accompanying source code is also available at <https://github.com/BinRoot/Haskell-Data-Analysis-Cookbook>.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [<service@packtpub.com>](mailto:service@packtpub.com) for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Preface

Data analysis is something that many of us have done before, maybe even without knowing it. It is the essential art of gathering and examining pieces of information to suit a variety of purposes—from visual inspection to machine learning techniques. Through data analysis, we can harness the meaning from information littered all around the digital realm. It enables us to resolve the most peculiar inquiries, perhaps even summoning new ones in the process.

Haskell acts as our conduit for robust data analysis. For some, Haskell is a programming language reserved to the most elite researchers in academia and industry. Yet, we see it as a charming one of the fastest growing cultures of open source developers around the world. The growth of Haskell is a sign that people are uncovering its magnificent functional pureness, resilient type safety, and remarkable expressiveness. Flip the pages of this book to see it all in action.

Haskell Data Analysis Cookbook is more than just a fusion of two entrancing topics in computing. It is also a learning tool for the Haskell programming language and an introduction to simple data analysis practices. Use it as a Swiss Army Knife of algorithms and code snippets. Try a recipe a day, like a kata for your mind. Breeze through the book for creative inspiration from catalytic examples. Also, most importantly, dive deep into the province of data analysis in Haskell.

Of course, none of this would have been possible without a thorough feedback from the technical editors, brilliant chapter illustrations by Lonku (<http://lonku.tumblr.com>), and helpful layout and editing support by Packt Publishing.

What this book covers

[Chapter 1](#), *The Hunt for Data*, identifies core approaches in reading data from various external sources such as CSV, JSON, XML, HTML, MongoDB, and SQLite.

[Chapter 2](#), *Integrity and Inspection*, explains the importance of cleaning data through recipes about trimming whitespaces, lexing, and regular expression matching.

[Chapter 3](#), *The Science of Words*, introduces common string manipulation algorithms, including base conversions, substring matching, and computing the edit distance.

[Chapter 4](#), *Data Hashing*, covers essential hashing functions such as MD5, SHA256, GeoHashing, and perceptual hashing.

[Chapter 5](#), *The Dance with Trees*, establishes an understanding of the tree data structure through examples that include tree traversals, balancing trees, and Huffman coding.

[Chapter 6](#), *Graph Fundamentals*, manifests rudimentary algorithms for graphical networks such as graph traversals, visualization, and maximal clique detection.

[Chapter 7](#), *Statistics and Analysis*, begins the investigation of important data analysis techniques that encompass regression algorithms, Bayesian networks, and neural networks.

[Chapter 8](#), *Clustering and Classification*, involves quintessential analysis methods that involve k-means clustering, hierarchical clustering, constructing decision trees, and implementing the k-Nearest Neighbors classifier.

[Chapter 9](#), *Parallel and Concurrent Design*, introduces advanced topics in Haskell such as forking I/O actions, mapping over lists in parallel, and benchmarking performance.

[Chapter 10](#), *Real-time Data*, incorporates streamed data interactions from Twitter, Internet Relay Chat (IRC), and sockets.

[Chapter 11](#), *Visualizing Data*, deals with sundry approaches to plotting graphs, including line charts, bar graphs, scatter plots, and [D3.js](#) visualizations.

[Chapter 12](#), *Exporting and Presenting*, concludes the book with an enumeration of algorithms for exporting data to CSV, JSON, HTML, MongoDB, and SQLite.

- [read online Men Without Women online](#)
- [We Can Build You pdf, azw \(kindle\)](#)
- [click Piece of Mind \(Emily the Strange, Book 4\) for free](#)
- [The Garden Party and Other Stories pdf, azw \(kindle\), epub](#)
- [download Mind Scrambler \(John Ceepak, Book 5\) book](#)
- [read online Ecclesiastes or, The Preacher](#)

- <http://metromekanik.com/ebooks/Anxious-Decades--America-in-Prosperity-and-Depression--1920-1941.pdf>
- <http://hasanetmekci.com/ebooks/Clouds-above-the-Hill--A-Historical-Novel-of-the-Russo-Japanese-War--Volume-2.pdf>
- <http://www.satilik-kopek.com/library/The-Caves-of-Alienation.pdf>
- <http://www.gateaerospaceforum.com/?library/Herbal-Supplements--Efficacy--Toxicity--Interactions-with-Western-Drugs--and-Effects-on-Clinical-Laboratory-Te>
- <http://tuscalaural.com/library/Unnatural-Death--Lord-Peter-Wimsey-Mysteries-.pdf>
- <http://test1.batsinbelfries.com/ebooks/Ecclesiastes-or--The-Preacher.pdf>