Geocomputation with R

Chapman & Hall/CRC
The R Series

Recently Published Titles

Spatial Microsimulation with R
Robin Lovelace, Morgane Dumont

Extending R
John M. Chambers

Using the R Commander: A Point-and-Click Interface for R
John Fox

Computational Actuarial Science with R
Arthur Charpentier

bookdown: Authoring Books and Technical Documents with R Markdown,
Yihui Xie

Testing R Code
Richard Cotton

R Primer, Second Edition
Claus Thorn Ekstrøm

Flexible Regression and Smoothing: Using GAMLSS in R
Mikis D. Stasinopoulos, Robert A. Rigby, Gillian Z. Heller, Vlasios Voudouris, and Fernanda De Bastiani

The Essentials of Data Science: Knowledge Discovery Using R
Graham J. Williams

blogdown: Creating Websites with R Markdown
Yihui Xie, Alison Presmanes Hill, Amber Thomas

Handbook of Educational Measurement and Psychometrics Using R
Christopher D. Desjardins, Okan Bulut

Displaying Time Series, Spatial, and Space-Time Data with R, Second Edition
Oscar Perpinan Lamigueiro

Reproducible Finance with R
Jonathan K. Regenstein, Jr

R Markdown
The Definitive Guide
Yihui Xie, J.J. Allaire, Garrett Grolemund

Practical R for Mass Communication and Journalism
Sharon Machlis

Analyzing Baseball Data with R, Second Edition
Max Marchi, Jim Albert, Benjamin S. Baumer

Spatio-Temporal Statistics with R
Christopher K. Wikle, Andrew Zammit-Mangion, and Noel Cressie

Statistical Computing with R, Second Edition
Maria L. Rizzo

Geocomputation with R
Robin Lovelace, Jakub Nowosad, Jannes Muenchow

For more information about this series, please visit: https://www.crcpress.com/go/the-r-series

Geocomputation with R

Robin Lovelace
Jakub Nowosad
Jannes Muenchow

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20190220

International Standard Book Number-13: 978-1-138-30451-2 (Hardback)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

For Katy

Dla Jagody

Für meine Katharina und alle unsere Kinder

Contents

Foreword

Doing ‘spatial’ in R has always been about being broad, seeking to provide and integrate tools from geography, geoinformatics, geocomputation and spatial statistics for anyone interested in joining in: joining in asking interesting questions, contributing fruitful research questions, and writing and improving code. That is, doing ‘spatial’ in R has always included open source code, open data and reproducibility.

Doing ‘spatial’ in R has also sought to be open to interaction with many branches of applied spatial data analysis, and also to implement new advances in data representation and methods of analysis to expose them to cross-disciplinary scrutiny. As this book demonstrates, there are often alternative workflows from similar data to similar results, and we may learn from comparisons with how others create and understand their workflows. This includes learning from similar communities around Open Source GIS and complementary languages such as Python, Java and so on.

R’s wide range of spatial capabilities would never have evolved without people willing to share what they were creating or adapting. This might include teaching materials, software, research practices (reproducible research, open data), and combinations of these. R users have also benefitted greatly from ‘upstream’ open source geo libraries such as GDAL, GEOS and PROJ.

This book is a clear example that, if you are curious and willing to join in, you can find things that need doing and that match your aptitudes. With advances in data representation and workflow alternatives, and ever increasing numbers of new users often without applied quantitative command-line exposure, a book of this kind has really been needed. Despite the effort involved, the authors have supported each other in pressing forward to publication.

So, this fresh book is ready to go; its authors have tried it out during many tutorials and workshops, so readers and instructors will be able to benefit from knowing that the contents have been and continue to be tried out on people like them. Engage with the authors and the wider R-spatial community, see value in having more choice in building your workflows and most important, enjoy applying what you learn here to things you care about.

Roger Bivand

Bergen, September 2018

Preface

This book is for people who want to analyze, visualize and model geographic data with open source software. It is based on R, a statistical programming language that has powerful data processing, visualization and geospatial capabilities. The book covers a wide range of topics and will be of interest to a wide range of people from many different backgrounds, especially:

- People who have learned spatial analysis skills using a desktop Geographic Information System (GIS) such as QGIS[1], ArcMap[2], GRASS[3] or SAGA[4], who want access to a powerful (geo)statistical and visualization programming language and the benefits of a command-line approach (Sherman, 2008):

> With the advent of 'modern' GIS software, most people want to point and click their way through life. That's good, but there is a tremendous amount of flexibility and power waiting for you with the command line.

- Graduate students and researchers from fields specializing in geographic data including Geography, Remote Sensing, Planning, GIS and Geographic Data Science
- Academics and post-graduate students working on projects in fields including Geology, Regional Science, Biology and Ecology, Agricultural Sciences (precision farming), Archaeology, Epidemiology, Transport Modeling, and broadly defined Data Science which require the power and flexibility of R for their research
- Applied researchers and analysts in public, private or third-sector organizations who need the reproducibility, speed and flexibility of a command-line language such as R in applications dealing with spatial data as diverse as

[1] `http://qgis.org/en/site/`
[2] `http://desktop.arcgis.com/en/arcmap/`
[3] `https://grass.osgeo.org/`
[4] `http://www.saga-gis.org/en/index.html`

Urban and Transport Planning, Logistics, Geo-marketing (store location analysis) and Emergency Planning

The book is designed for intermediate-to-advanced R users interested in geocomputation and R beginners who have prior experience with geographic data. If you are new to both R and geographic data, do not be discouraged: we provide links to further materials and describe the nature of spatial data from a beginner's perspective in Chapter 2 and in links provided below.

How to read this book

The book is divided into three parts:

1. Part I: Foundations, aimed at getting you up-to-speed with geographic data in R.
2. Part II: Extensions, which covers advanced techniques.
3. Part III: Applications, to real-world problems.

The chapters get progressively harder in each so we recommend reading the book in order. A major barrier to geographical analysis in R is its steep learning curve. The chapters in Part I aim to address this by providing reproducible code on simple datasets that should ease the process of getting started.

An important aspect of the book from a teaching/learning perspective is the **exercises** at the end of each chapter. Completing these will develop your skills and equip you with the confidence needed to tackle a range of geospatial problems. Solutions to the exercises, and a number of extended examples, are provided on the book's supporting website, at geocompr.github.io[5].

Impatient readers are welcome to dive straight into the practical examples, starting in Chapter 2. However, we recommend reading about the wider context of *Geocomputation with R* in Chapter 1 first. If you are new to R, we also recommend learning more about the language before attempting to run the code chunks provided in each chapter (unless you're reading the book for an understanding of the concepts). Fortunately for R beginners R has a supportive community that has developed a wealth of resources that can help. We particularly recommend three tutorials: R for Data Science[6] (Grolemund and Wickham, 2016) and Efficient R Programming[7] (Gillespie and Lovelace, 2016), especially Chapter 2[8] (on installing and setting-up R/RStudio) and

[5] `https://geocompr.github.io/`
[6] `http://r4ds.had.co.nz/`
[7] `https://csgillespie.github.io/efficientR/`
[8] `https://csgillespie.github.io/efficientR/set-up.html#r-version`

Chapter 10[9] (on learning to learn), and An introduction to R[10] (Venables et al., 2017). A good interactive tutorial is DataCamp's Introduction to R[11].

Why R?

Although R has a steep learning curve, the command-line approach advocated in this book can quickly pay off. As you'll learn in subsequent chapters, R is an effective tool for tackling a wide range of geographic data challenges. We expect that, with practice, R will become the program of choice in your geospatial toolbox for many applications. Typing and executing commands at the command-line is, in many cases, faster than pointing-and-clicking around the graphical user interface (GUI) a desktop GIS. For some applications such as Spatial Statistics and modeling R may be the *only* realistic way to get the work done.

As outlined in Section 1.2, there are many reasons for using R for geocomputation: R is well-suited to the interactive use required in many geographic data analysis workflows compared with other languages. R excels in the rapidly growing fields of Data Science (which includes data carpentry, statistical learning techniques and data visualization) and Big Data (via efficient interfaces to databases and distributed computing systems). Furthermore R enables a reproducible workflow: sharing scripts underlying your analysis will allow others to build-on your work. To ensure reproducibility in this book we have made its source code available at github.com/Robinlovelace/geocompr[12]. There you will find script files in the `code/` folder that generate figures: when code generating a figure is not provided in the main text of the book, the name of the script file that generated it is provided in the caption (see for example the caption for Figure 12.2).

Other languages such as Python, Java and C++ can be used for geocomputation and there are excellent resources for learning geocomputation *without R*, as discussed in Section 1.3. None of these provide the unique combination of package ecosystem, statistical capabilities, visualization options, powerful IDEs offered by the R community. Furthermore, by teaching how to use one language (R) in depth, this book will equip you with the concepts and confidence needed to do geocomputation in other languages.

Geocomputation with R will equip you with knowledge and skills to tackle a wide range of issues, including those with scientific, societal and environmental

[9] `https://csgillespie.github.io/efficientR/learning.html`
[10] `http://colinfay.me/intro-to-r/`
[11] `https://www.datacamp.com/courses/free-introduction-to-r`
[12] `https://github.com/Robinlovelace/geocompr#geocomputation-with-r`

implications, manifested in geographic data. As described in Section 1.1, geocomputation is not only about using computers to process geographic data: it is also about real-world impact. If you are interested in the wider context and motivations behind this book, read on; these are covered in Chapter 1.

Acknowledgements

Many thanks to everyone who contributed directly and indirectly via the code hosting and collaboration site GitHub, including the following people who contributed direct via pull requests: katygregg, erstearns, eyesofbambi, tyluRp, marcosci, mdsumner, rsbivand, pat-s, gisma, ateucher, annakrystalli, gavin-simpson, Himanshuteli, yutannihilation, katiejolly, layik, mvl22, nickbearman, ganes1410, richfitz, SymbolixAU, wdearden, yihui, chihinl, gregor-d, p-kono, pokyah. Special thanks to Marco Sciaini, who not only created the front cover image, but also published the code that generated it (see `frontcover.R` in the book's GitHub repo). Dozens more people contributed online, by raising and commenting on issues, and by providing feedback via social media. The `#geocompr` hashtag will live on!

We would like to thank John Kimmel from CRC Press, who has worked with us over two years to take our ideas from an early book plan into production via four rounds of peer review. The reviewers deserve special mention here: their detailed feedback and expertise substantially improved the book's structure and content.

We thank Patrick Schratz and Alexander Brenning from the University of Jena for fruitful discussions on and input into Chapters 11 and 14. We thank Emmanuel Blondel from the Food and Agriculture Organization of the United Nations for expert input into the section on web services; Michael Sumner for critical input into many areas of the book, especially the discussion of algorithms in Chapter 10; Tim Appelhans and David Cooley for key contributions to the visualization chapter (Chapter 8); and Katy Gregg, who proofread every chapter and greatly improved the readability of the book.

Countless others could be mentioned who contributed in myriad ways. The final thank you is for all the software developers who make geocomputation with R possible. Edzer Pebesma (who created the **sf** package), Robert Hijmans (who created **raster**) and Roger Bivand (who laid the foundations for much R-spatial software) have made high performance geographic computing possible in R.

1

Introduction

This book is about using the power of computers to *do things* with geographic data. It teaches a range of spatial skills, including: reading, writing and manipulating geographic data; making static and interactive maps; applying geocomputation to solve real-world problems; and modeling geographic phenomena. By demonstrating how various geographic operations can be linked, in reproducible 'code chunks' that intersperse the prose, the book also teaches a transparent and thus scientific workflow. Learning how to use the wealth of geospatial tools available from the R command line can be exciting, but creating *new ones* can be truly liberating. Using the command-line driven approach taught throughout, and programming techniques covered in Chapter 10, can help remove constraints on your creativity imposed by software. After reading the book and completing the exercises, you should therefore feel empowered with a strong understanding of the possibilities opened up by R's impressive geographic capabilities, new skills to solve real-world problems with geographic data, and the ability to communicate your work with maps and reproducible code.

Over the last few decades free and open source software for geospatial (FOSS4G) has progressed at an astonishing rate. Thanks to organizations such as OSGeo, geographic data analysis is no longer the preserve of those with expensive hardware and software: anyone can now download and run high-performance spatial libraries. Open source Geographic Information Systems (GIS) such as QGIS[1] have made geographic analysis accessible worldwide. GIS programs tend to emphasize graphical user interfaces (GUIs), with the unintended consequence of discouraging reproducibility (although many can be used from the command line as we'll see in Chapter 9). R, by contrast, emphasizes the command line interface (CLI). A simplistic comparison between the different approaches is illustrated in Table 1.1.

This book is motivated by the importance of reproducibility for scientific research (see the note below). It aims to make reproducible geographic data analysis workflows more accessible, and demonstrate the power of open geospatial software available from the command-line. "Interfaces to other software are part of R" (Eddelbuettel and Balamuta, 2018). This means that in addition to outstanding 'in house' capabilities, R allows access to many other spatial

[1] `http://qgis.org/en/site/`

TABLE 1.1: Differences in emphasis between software packages (Graphical User Interface (GUI) of Geographic Information Systems (GIS) and R).

Attribute	Desktop GIS (GUI)	R
Home disciplines	Geography	Computing, Statistics
Software focus	Graphical User Interface	Command line
Reproducibility	Minimal	Maximal

software libraries, explained in Section 1.2 and demonstrated in Chapter 9. Before going into the details of the software, however, it is worth taking a step back and thinking about what we mean by geocomputation.

Reproducibility is a major advantage of command-line interfaces, but what does it mean in practice? We define it as follows: "A process in which the same results can be generated by others using publicly accessible code."

This may sound simple and easy to achieve (which it is if you carefully maintain your R code in script files), but has profound implications for teaching and the scientific process (Pebesma et al., 2012).

1.1 What is geocomputation?

Geocomputation is a young term, dating back to the first conference on the subject in 1996.[2] What distinguished geocomputation from the (at the time) commonly used term 'quantitative geography', its early advocates proposed, was its emphasis on "creative and experimental" applications (Longley et al., 1998) and the development of new tools and methods (Openshaw and Abrahart, 2000): "GeoComputation is about using the various different types of geodata and about developing relevant geo-tools within the overall context of a 'scientific' approach." This book aims to go beyond teaching methods and code; by the end of it you should be able to use your geocomputational skills, to do "practical work that is beneficial or useful" (Openshaw and Abrahart, 2000).

Our approach differs from early adopters such as Stan Openshaw, however, in its emphasis on reproducibility and collaboration. At the turn of the 21st Century, it was unrealistic to expect readers to be able to reproduce code

[2]The conference took place at the University of Leeds, where one of the authors (Robin) is currently based. The 21st GeoComputation conference was also hosted at the University of Leeds, during which Robin and Jakub presented, led a workshop on 'tidy' spatial data analysis and collaborated on the book (see www.geocomputation.org for more on the conference series, and papers/presentations spanning two decades).

examples, due to barriers preventing access to the necessary hardware, software and data. Fast-forward two decades and things have progressed rapidly. Anyone with access to a laptop with ~4GB RAM can realistically expect to be able to install and run software for geocomputation on publicly accessible datasets, which are more widely available than ever before (as we will see in Chapter 7).[3] Unlike early works in the field, all the work presented in this book is reproducible using code and example data supplied alongside the book, in R packages such as **spData**, the installation of which is covered in Chapter 2.

Geocomputation is closely related to other terms including: Geographic Information Science (GIScience); Geomatics; Geoinformatics; Spatial Information Science; Geoinformation Engineering (Longley, 2015); and Geographic Data Science (GDS). Each term shares an emphasis on a 'scientific' (implying reproducible and falsifiable) approach influenced by GIS, although their origins and main fields of application differ. GDS, for example, emphasizes 'data science' skills and large datasets, while Geoinformatics tends to focus on data structures. But the overlaps between the terms are larger than the differences between them and we use geocomputation as a rough synonym encapsulating all of them: they all seek to use geographic data for applied scientific work. Unlike early users of the term, however, we do not seek to imply that there is any cohesive academic field called 'Geocomputation' (or 'GeoComputation' as Stan Openshaw called it). Instead, we define the term as follows: working with geographic data in a computational way, focusing on code, reproducibility and modularity.

Geocomputation is a recent term but is influenced by old ideas. It can be seen as a part of Geography, which has a 2000+ year history (Talbert, 2014); and an extension of *Geographic Information Systems* (GIS) (Neteler and Mitasova, 2008), which emerged in the 1960s (Coppock and Rhind, 1991).

Geography has played an important role in explaining and influencing humanity's relationship with the natural world long before the invention of the computer, however. Alexander von Humboldt's travels to South America in the early 1800s illustrates this role: not only did the resulting observations lay the foundations for the traditions of physical and plant geography, they also paved the way towards policies to protect the natural world (Wulf, 2015). This book aims to contribute to the 'Geographic Tradition' (Livingstone, 1992) by harnessing the power of modern computers and open source software.

The book's links to older disciplines were reflected in suggested titles for the book: *Geography with R* and *R for GIS*. Each has advantages. The former

[3] A laptop with 4GB running a modern operating system such as Ubuntu 16.04 onward should also be able to reproduce the contents of this book. A laptop with this specification or above can be acquired second-hand for ~US$100 in many countries nowadays, reducing the financial/hardware barrier to geocomputation far below the levels in operation in the early 2000s, when high-performance computers were unaffordable for most people.

conveys the message that it comprises much more than just spatial data: non-spatial attribute data are inevitably interwoven with geometry data, and Geography is about more than where something is on the map. The latter communicates that this is a book about using R as a GIS, to perform spatial operations on *geographic data* (Bivand et al., 2013). However, the term GIS conveys some connotations (see Table 1.1) which simply fail to communicate one of R's greatest strengths: its console-based ability to seamlessly switch between geographic and non-geographic data processing, modeling and visualization tasks. By contrast, the term geocomputation implies reproducible and creative programming. Of course, (geocomputational) algorithms are powerful tools that can become highly complex. However, all algorithms are composed of smaller parts. By teaching you its foundations and underlying structure, we aim to empower you to create your own innovative solutions to geographic data problems.

1.2 Why use R for geocomputation?

Early geographers used a variety of tools including barometers, compasses and sextants[4] to advance knowledge about the world (Wulf, 2015). It was only with the invention of the marine chronometer[5] in 1761 that it became possible to calculate longitude at sea, enabling ships to take more direct routes.

Nowadays such lack of geographic data is hard to imagine. Every smartphone has a global positioning (GPS) receiver and a multitude of sensors on devices ranging from satellites and semi-autonomous vehicles to citizen scientists incessantly measure every part of the world. The rate of data produced is overwhelming. An autonomous vehicle, for example, can generate 100 GB of data per day (The Economist, 2016). Remote sensing data from satellites has become too large to analyze the corresponding data with a single computer, leading to initiatives such as OpenEO[6].

This 'geodata revolution' drives demand for high performance computer hardware and efficient, scalable software to handle and extract signal from the noise, to understand and perhaps change the world. Spatial databases enable storage and generation of manageable subsets from the vast geographic data stores, making interfaces for gaining knowledge from them vital tools for the future. R is one such tool, with advanced analysis, modeling and visualization capabilities. In this context the focus of the book is not on the language itself (see Wickham, 2014a). Instead we use R as a 'tool for the trade' for

[4] https://en.wikipedia.org/wiki/Sextant

[5] https://en.wikipedia.org/wiki/Marine_chronometer

[6] http://r-spatial.org/2016/11/29/openeo.html

understanding the world, similar to Humboldt's use of tools to gain a deep understanding of nature in all its complexity and interconnections (see Wulf, 2015). Although programming can seem like a reductionist activity, the aim is to teach geocomputation with R not only for fun, but for understanding the world.

R is a multi-platform, open source language and environment for statistical computing and graphics (r-project.org/[7]). With a wide range of packages, R also supports advanced geospatial statistics, modeling and visualization. New integrated development environments (IDEs) such as RStudio have made R more user-friendly for many, easing map making with a panel dedicated to interactive visualization.

At its core, R is an object-oriented, functional programming language[8] (Wickham, 2014a), and was specifically designed as an interactive interface to other software (Chambers, 2016). The latter also includes many 'bridges' to a treasure trove of GIS software, 'geolibraries' and functions (see Chapter 9). It is thus ideal for quickly creating 'geo-tools', without needing to master lower level languages (compared to R) such as C, FORTRAN or Java (see Section 1.3). This can feel like breaking free from the metaphorical 'glass ceiling' imposed by GUI-based or proprietary geographic information systems (see Table 1.1 for a definition of GUI). Furthermore, R facilitates access to other languages: the packages **Rcpp** and **reticulate** enable access to C++ and Python code, for example. This means R can be used as a 'bridge' to a wide range of geospatial programs (see Section 1.3).

Another example showing R's flexibility and evolving geographic capabilities is interactive map making. As we'll see in Chapter 8, the statement that R has "limited interactive [plotting] facilities" (Bivand et al., 2013) is no longer true. This is demonstrated by the following code chunk, which creates Figure 1.1 (the functions that generate the plot are covered in Section 8.4).

```
library(leaflet)
popup = c("Robin", "Jakub", "Jannes")
leaflet() %>%
  addProviderTiles("NASAGIBS.ViirsEarthAtNight2012") %>%
  addMarkers(lng = c(-3, 23, 11),
             lat = c(52, 53, 49),
             popup = popup)
```

It would have been difficult to produce Figure 1.1 using R a few years ago, let alone as an interactive map. This illustrates R's flexibility and how, thanks to developments such as **knitr** and **leaflet**, it can be used as an interface to other software, a theme that will recur throughout this book. The use of

[7] https://www.r-project.org/

[8] http://adv-r.had.co.nz/Functional-programming.html

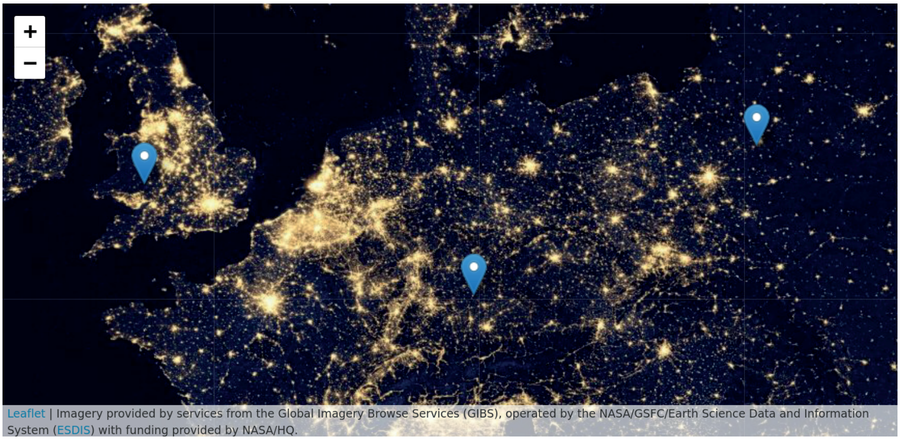

FIGURE 1.1: The blue markers indicate where the authors are from. The basemap is a tiled image of the Earth at night provided by NASA. Interact with the online version at geocompr.robinlovelace.net, for example by zooming in and clicking on the popups.

R code, therefore, enables teaching geocomputation with reference to reproducible examples such as that provided in Figure 1.1 rather than abstract concepts.

1.3 Software for geocomputation

R is a powerful language for geocomputation but there are many other options for geographic data analysis providing thousands of geographic functions. Awareness of other languages for geocomputation will help decide when a different tool may be more appropriate for a specific task, and place R in the wider geospatial ecosystem. This section briefly introduces the languages C++[9], Java[10] and Python[11] for geocomputation, in preparation for Chapter 9.

An important feature of R (and Python) is that it is an interpreted language. This is advantageous because it enables interactive programming in a Read–Eval–Print Loop (REPL): code entered into the console is immediately executed and the result is printed, rather than waiting for the intermediate

[9] `https://isocpp.org/`
[10] `https://www.oracle.com/java/index.html`
[11] `https://www.python.org/`

stage of compilation. On the other hand, compiled languages such as C++ and Java tend to run faster (once they have been compiled).

C++ provides the basis for many GIS packages such as QGIS[12], GRASS[13] and SAGA[14] so it is a sensible starting point. Well-written C++ is very fast, making it a good choice for performance-critical applications such as processing large geographic datasets, but is harder to learn than Python or R. C++ has become more accessible with the **Rcpp** package, which provides a good 'way in' to C programming for R users. Proficiency with such low-level languages opens the possibility of creating new, high-performance 'geoalgorithms' and a better understanding of how GIS software works (see Chapter 10).

Java is another important and versatile language for geocomputation. GIS packages gvSig, OpenJump and uDig are all written in Java. There are many GIS libraries written in Java, including GeoTools and JTS, the Java Topology Suite (GEOS is a C++ port of JTS). Furthermore, many map server applications use Java including Geoserver/Geonode, deegree and 52°North WPS.

Java's object-oriented syntax is similar to that of C++. A major advantage of Java is that it is platform-independent (which is unusual for a compiled language) and is highly scalable, making it a suitable language for IDEs such as RStudio, with which this book was written. Java has fewer tools for statistical modeling and visualization than Python or R, although it can be used for data science (Brzustowicz, 2017).

Python is an important language for geocomputation especially because many Desktop GIS such as GRASS, SAGA and QGIS provide a Python API (see Chapter 9). Like R, it is a popular[15] tool for data science. Both languages are object-oriented, and have many areas of overlap, leading to initiatives such as the **reticulate** package that facilitates access to Python from R and the Ursa Labs[16] initiative to support portable libraries to the benefit of the entire open source data science ecosystem.

In practice both R and Python have their strengths and to some extent which you use is less important than the domain of application and communication of results. Learning either will provide a head-start in learning the other. However, there are major advantages of R over Python for geocomputation. This includes its much better support of the geographic data models vector and raster in the language itself (see Chapter 2) and corresponding visualization possibilities (see Chapters 2 and 8). Equally important, R has unparalleled support for statistics, including spatial statistics, with hundreds of packages (unmatched by Python) supporting thousands of statistical methods.

[12] `www.qgis.org`

[13] `https://grass.osgeo.org/`

[14] `www.saga-gis.org`

[15] `https://stackoverflow.blog/2017/10/10/impressive-growth-r/`

[16] `https://ursalabs.org/`

The major advantage of Python is that it is a *general-purpose* programming language. It is used in many domains, including desktop software, computer games, websites and data science. Python is often the only shared language between different (geocomputation) communities and can be seen as the 'glue' that holds many GIS programs together. Many geoalgorithms, including those in QGIS and ArcMap, can be accessed from the Python command line, making it well-suited as a starter language for command-line GIS.[17]

For spatial statistics and predictive modeling, however, R is second-to-none. This does not mean you must choose either R or Python: Python supports most common statistical techniques (though R tends to support new developments in spatial statistics earlier) and many concepts learned from Python can be applied to the R world. Like R, Python also supports geographic data analysis and manipulation with packages such as **osgeo**, **Shapely**, **NumPy** and **PyGeoProcessing** (Garrard, 2016).

1.4 R's spatial ecosystem

There are many ways to handle geographic data in R, with dozens of packages in the area.[18] In this book we endeavor to teach the state-of-the-art in the field whilst ensuring that the methods are future-proof. Like many areas of software development, R's spatial ecosystem is rapidly evolving (Figure 1.2). Because R is open source, these developments can easily build on previous work, by 'standing on the shoulders of giants', as Isaac Newton put it in 1675[19]. This approach is advantageous because it encourages collaboration and avoids 'reinventing the wheel'. The package **sf** (covered in Chapter 2), for example, builds on its predecessor **sp**.

A surge in development time (and interest) in 'R-spatial' has followed the award of a grant by the R Consortium for the development of support for Simple Features, an open-source standard and model to store and access vector geometries. This resulted in the **sf** package (covered in Section 2.2.1). Multiple places reflect the immense interest in **sf**. This is especially true for the R-sig-Geo Archives[20], a long-standing open access email list containing much R-spatial wisdom accumulated over the years.

It is noteworthy that shifts in the wider R community, as exemplified by the data

[17] Python modules providing access to geoalgorithms include `grass.script` for GRASS, `saga-python` for SAGA-GIS, `processing` for QGIS and `arcpy` for ArcGIS.

[18] An overview of R's spatial ecosystem can be found in the CRAN Task View on the Analysis of Spatial Data (see `https://cran.r-project.org/web/views/Spatial.html`).

[19] `http://digitallibrary.hsp.org/index.php/Detail/Object/Show/object_id/9285`

[20] `https://stat.ethz.ch/pipermail/r-sig-geo/`

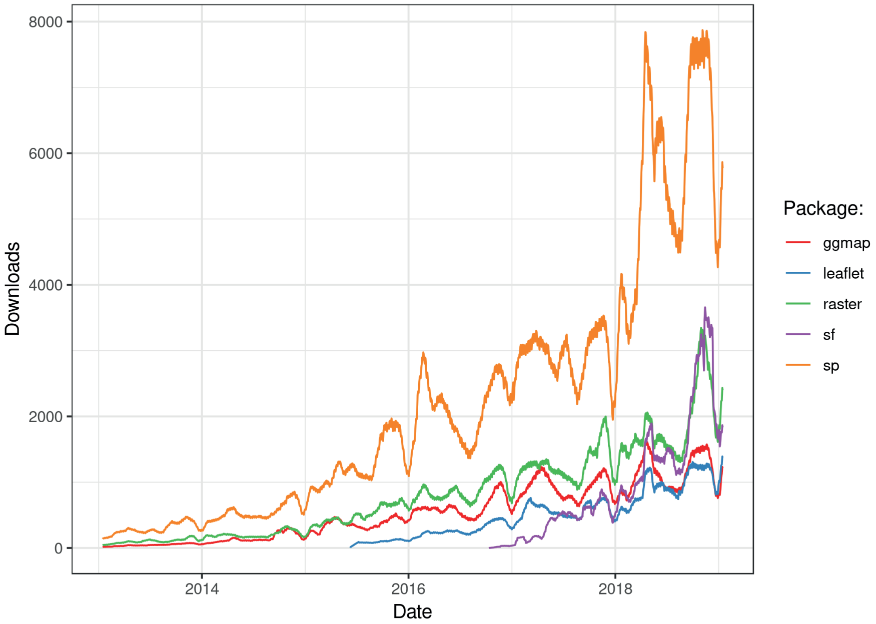

FIGURE 1.2: The popularity of spatial packages in R. The y-axis shows average number of downloads per day, within a 30-day rolling window, of prominent spatial packages.

processing package **dplyr** (released in 2014[21]) influenced shifts in R's spatial ecosystem. Alongside other packages that have a shared style and emphasis on 'tidy data' (including, e.g., **ggplot2**), **dplyr** was placed in the **tidyverse** 'metapackage' in late 2016[22]. The **tidyverse** approach, with its focus on long-form data and fast intuitively named functions, has become immensely popular. This has led to a demand for 'tidy geographic data' which has been partly met by **sf** and other approaches such as **tabularaster**. An obvious feature of the **tidyverse** is the tendency for packages to work in harmony. There is no equivalent **geoverse**, but there are attempts at harmonization between packages hosted in the r-spatial[23] organization and a growing number of packages use **sf** (Table 1.2).

[21] `https://cran.r-project.org/src/contrib/Archive/dplyr/`
[22] `https://cran.r-project.org/src/contrib/Archive/tidyverse/`
[23] `https://github.com/r-spatial/discuss/issues/11`

TABLE 1.2: The top 5 most downloaded packages that depend on sf, in terms of average number of downloads per day over the previous month. As of 2019-01-29 there are 117 packages which import sf.

package	Downloads
ggplot2	22593
plotly	3628
raster	2433
leaflet	1391
spData	1174

1.5 The history of R-spatial

There are many benefits of using recent spatial packages such as **sf**, but it also important to be aware of the history of R's spatial capabilities: many functions, use-cases and teaching material are contained in older packages. These can still be useful today, provided you know where to look.

R's spatial capabilities originated in early spatial packages in the S language (Bivand and Gebhardt, 2000). The 1990s saw the development of numerous S scripts and a handful of packages for spatial statistics. R packages arose from these and by 2000 there were R packages for various spatial methods "point pattern analysis, geostatistics, exploratory spatial data analysis and spatial econometrics", according to an article[24] presented at GeoComputation 2000 (Bivand and Neteler, 2000). Some of these, notably **spatial**, **sgeostat** and **splancs** are still available on CRAN (Rowlingson and Diggle, 1993, 2017; Venables and Ripley, 2002; Majure and Gebhardt, 2016).

A subsequent article in R News (the predecessor of The R Journal[25]) contained an overview of spatial statistical software in R at the time, much of which was based on previous code written for S/S-PLUS (Ripley, 2001). This overview described packages for spatial smoothing and interpolation, including **akima** and **geoR** (Akima and Gebhardt, 2016; Jr and Diggle, 2016), and point pattern analysis, including **splancs** (Rowlingson and Diggle, 2017) and **spatstat** (Baddeley et al., 2015).

The following R News issue (Volume 1/3) put spatial packages in the spotlight again, with a more detailed introduction to **splancs** and a commentary on future prospects regarding spatial statistics (Bivand, 2001). Additionally, the issue introduced two packages for testing spatial autocorrelation that eventually became part of **spdep** (Bivand, 2017). Notably, the commentary mentions

[24] `http://www.geocomputation.org/2000/GC009/Gc009.htm`
[25] `https://journal.r-project.org/`

the need for standardization of spatial interfaces, efficient mechanisms for exchanging data with GIS, and handling of spatial metadata such as coordinate reference systems (CRS).

maptools (written by Nicholas Lewin-Koh; Bivand and Lewin-Koh, 2017) is another important package from this time. Initially **maptools** just contained a wrapper around shapelib[26] and permitted the reading of ESRI Shapefiles into geometry nested lists. The corresponding and nowadays obsolete S3 class called "Map" stored this list alongside an attribute data frame. The work on the "Map" class representation was nevertheless important since it directly fed into **sp** prior to its publication on CRAN.

In 2003 Roger Bivand published an extended review of spatial packages. It proposed a class system to support the "data objects offered by GDAL", including 'fundamental' point, line, polygon, and raster types. Furthermore, it suggested interfaces to external libraries should form the basis of modular R packages (Bivand, 2003). To a large extent these ideas were realized in the packages **rgdal** and **sp**. These provided a foundation for spatial data analysis with R, as described in *Applied Spatial Data Analysis with R* (ASDAR) (Bivand et al., 2013), first published in 2008. Ten years later, R's spatial capabilities have evolved substantially but they still build on ideas set-out by Bivand (2003): interfaces to GDAL and PROJ, for example, still power R's high-performance geographic data I/O and CRS transformation capabilities (see Chapters 6 and 7, respectively).

rgdal, released in 2003, provided GDAL bindings for R which greatly enhanced its ability to import data from previously unavailable geographic data formats. The initial release supported only raster drivers but subsequent enhancements provided support for coordinate reference systems (via the PROJ library), reprojections and import of vector file formats (see Chapter 7 for more on file formats). Many of these additional capabilities were developed by Barry Rowlingson and released in the **rgdal** codebase in 2006 (see Rowlingson et al., 2003, and the R-help[27] email list for context).

sp, released in 2005, overcame R's inability to distinguish spatial and non-spatial objects (Pebesma and Bivand, 2005). **sp** grew from a workshop[28] in Vienna in 2003 and was hosted at sourceforge before migrating to R-Forge[29]. Prior to 2005, geographic coordinates were generally treated like any other number. **sp** changed this with its classes and generic methods supporting points, lines, polygons and grids, and attribute data.

sp stores information such as bounding box, coordinate reference system and attributes in slots in `Spatial` objects using the S4 class system, enabling data operations to work on geographic data (see Section 2.2.2). Further, **sp**

[26] `http://shapelib.maptools.org/`

[27] `https://stat.ethz.ch/pipermail/r-help/2003-January/028413.html`

[28] `http://spatial.nhh.no/meetings/vienna/index.html`

[29] `https://r-forge.r-project.org`

provides generic methods such as `summary()` and `plot()` for geographic data. In the following decade, **sp** classes rapidly became popular for geographic data in R and the number of packages that depended on it increased from around 20 in 2008 to over 100 in 2013 (Bivand et al., 2013). As of 2018 almost 500 packages rely on **sp**, making it an important part of the R ecosystem. Prominent R packages using **sp** include: **gstat**, for spatial and spatio-temporal geostatistics; **geosphere**, for spherical trigonometry; and **adehabitat** used for the analysis of habitat selection by animals (Pebesma and Graeler, 2018; Calenge, 2006; Hijmans, 2016).

While **rgdal** and **sp** solved many spatial issues, R still lacked the ability to do geometric operations (see Chapter 5). Colin Rundel addressed this issue by developing **rgeos**, an R interface to the open-source geometry library (GEOS) during a Google Summer of Code project in 2010 (Bivand and Rundel, 2018). **rgeos** enabled GEOS to manipulate **sp** objects, with functions such as `gIntersection()`.

Another limitation of **sp** — its limited support for raster data — was overcome by **raster**, first released in 2010 (Hijmans, 2017). Its class system and functions support a range of raster operations as outlined in Section 2.3. A key feature of **raster** is its ability to work with datasets that are too large to fit into RAM (R's interface to PostGIS supports off-disc operations on vector geographic data). **raster** also supports map algebra (see Section 4.3.2).

In parallel with these developments of class systems and methods came the support for R as an interface to dedicated GIS software. **GRASS** (Bivand, 2000) and follow-on packages **spgrass6** and **rgrass7** (for GRASS GIS 6 and 7, respectively) were prominent examples in this direction (Bivand, 2016a,b). Other examples of bridges between R and GIS include **RSAGA** (Brenning et al., 2018, first published in 2008), **RPyGeo** (Brenning, 2012a, first published in 2008), and **RQGIS** (Muenchow et al., 2017, first published in 2016) (see Chapter 9).

Visualization was not a focus initially, with the bulk of R-spatial development focused on analysis and geographic operations. **sp** provided methods for map making using both the base and lattice plotting system but demand was growing for advanced map making capabilities, especially after the release of **ggplot2** in 2007. **ggmap** extended **ggplot2**'s spatial capabilities (Kahle and Wickham, 2013), by facilitating access to 'basemap' tiles from online services such as Google Maps. Though **ggmap** facilitated map-making with **ggplot2**, its utility was limited by the need to `fortify` spatial objects, which means converting them into long data frames. While this works well for points it is computationally inefficient for lines and polygons, since each coordinate (vertex) is converted into a row, leading to huge data frames to represent complex geometries. Although geographic visualization tended to focus on vector data, raster visualization is supported in **raster** and received a boost with the release of **rasterVis**, which is described in a book on the subject of spatial and temporal data visualization (Lamigueiro, 2018). As of 2018 map making in R is a hot topic with dedicated packages such as **tmap**, **leaflet** and **mapview** all supporting the class system provided by **sf**, the focus of the next chapter (see Chapter 8 for more on visualization).

1.6 Exercises

1. Think about the terms 'GIS', 'GDS' and 'geocomputation' described above. Which (if any) best describes the work you would like to do using geo* methods and software and why?
2. Provide three reasons for using a scriptable language such as R for geocomputation instead of using an established GIS program such as QGIS.
3. Name two advantages and two disadvantages of using mature vs recent packages for geographic data analysis (for example **sp** vs **sf**).

Bibliography

Abelson, H., Sussman, G. J., and Sussman, J. (1996). *Structure and Interpretation of Computer Programs*. The MIT Electrical Engineering and Computer Science Series. MIT Press, Cambridge, Massachusetts, second edition.

Akima, H. and Gebhardt, A. (2016). *Akima: Interpolation of Irregularly and Regularly Spaced Data*.

Baddeley, A., Rubak, E., and Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. CRC Press.

Baddeley, A. and Turner, R. (2005). Spatstat: An R package for analyzing spatial point patterns. *Journal of statistical software*, 12(6):1–42.

Bellos, A. (2011). *Alex's Adventures in Numberland*. Bloomsbury Paperbacks, London.

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., and Jones, Z. M. (2016). Mlr: Machine Learning in R. *Journal of Machine Learning Research*, 17(170):1–5.

Bivand, R. (2001). More on Spatial Data Analysis. *R News*, 1(3):13–17.

Bivand, R. (2003). Approaches to Classes for Spatial Data in R. In Hornik, K., Leisch, F., and Zeileis, A., editors, *Proceedings of DSC*.

Bivand, R. (2016a). *Rgrass7: Interface Between GRASS 7 Geographical Information System and R*.

Bivand, R. (2016b). *Spgrass6: Interface between GRASS 6 and R*.

Bivand, R. (2017). *Spdep: Spatial Dependence: Weighting Schemes, Statistics and Models*.

Bivand, R. and Gebhardt, A. (2000). Implementing functions for spatial statistical analysis using the language. *Journal of Geographical Systems*, 2(3):307–317.

Bivand, R., Keitt, T., and Rowlingson, B. (2018). *rgdal: Bindings for the 'Geospatial' Data Abstraction Library*. R package version 1.3-3.

Bivand, R. and Lewin-Koh, N. (2017). *Maptools: Tools for Reading and Handling Spatial Objects*.

Bivand, R. and Neteler, M. (2000). Open source geocomputation: Using the R data analysis language integrated with GRASS GIS and PostgreSQL data base systems. In Neteler, M. and Bivand, R. S., editors, *Proceedings of the 5th International Conference on GeoComputation.*

Bivand, R., Pebesma, E. J., and Gómez-Rubio, V. (2013). *Applied Spatial Data Analysis with R*, volume 747248717. Springer.

Bivand, R. and Rundel, C. (2018). *rgeos: Interface to Geometry Engine - Open Source ('GEOS').* R package version 0.3-28.

Bivand, R. S. (2000). Using the R statistical data analysis language on GRASS 5.0 GIS database files. *Computers & Geosciences*, 26(9):1043–1052.

Blangiardo, M. and Cameletti, M. (2015). *Spatial and Spatio-Temporal Bayesian Models with R-INLA.* John Wiley & Sons, Ltd, Chichester, UK.

Borcard, D., Gillet, F., and Legendre, P. (2011). *Numerical Ecology with R.* Use R! Springer, New York. OCLC: ocn690089213.

Borland, D. and Taylor II, R. M. (2007). Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2).

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Brenning, A. (2012a). *ArcGIS Geoprocessing in R via Python.*

Brenning, A. (2012b). Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The R package sperrorest. pages 5372–5375. IEEE.

Brenning, A., Bangs, D., and Becker, M. (2018). *RSAGA: SAGA Geoprocessing and Terrain Analysis.* R package version 1.1.0.

Brewer, C. A. (2015). *Designing Better Maps: A Guide for GIS Users.* Esri Press, Redlands, California, second edition.

Bristol City Council (2015). Deprivation in Bristol 2015. Technical report, Bristol City Council.

Brus, D. J. (2018). Sampling for digital soil mapping: A tutorial supported by R scripts. *Geoderma.*

Brzustowicz, M. R. (2017). *Data Science with Java: [Practical Methods for Scientists and Engineers].* O´Reilly, Beijing Boston Farnham, first edition. OCLC: 993428657.

Bucklin, D. and Basille, M. (2018). Rpostgis: Linking R with a PostGIS Spatial Database. *The R Journal.*

Burrough, P. A., McDonnell, R., and Lloyd, C. D. (2015). *Principles of Geographical Information Systems.* Oxford University Press, Oxford, New York, third edition. OCLC: ocn915100245.

Calenge, C. (2006). The package adehabitat for the R software: Tool for the analysis of space and habitat use by animals. *Ecological Modelling*, 197:1035.

Cawley, G. C. and Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107.

Chambers, J. M. (2016). *Extending R.* CRC Press.

Cheshire, J. and Lovelace, R. (2015). Spatial data visualisation with R. In Brunsdon, C. and Singleton, A., editors, *Geocomputation*, pages 1–14. SAGE Publications.

Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V., and Böhner, J. (2015). System for Automated Geoscientific Analyses (SAGA) v. 2.1.4. *Geosci. Model Dev.*, 8(7):1991–2007.

Coombes, M. G., Green, A. E., and Openshaw, S. (1986). An Efficient Algorithm to Generate Official Statistical Reporting Areas: The Case of the 1984 Travel-to-Work Areas Revision in Britain. *The Journal of the Operational Research Society*, 37(10):943.

Coppock, J. T. and Rhind, D. W. (1991). The history of GIS. *Geographical Information Systems: Principles and Applications, vol. 1.*, 1(1):21–43.

de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications.* Springer Science & Business Media.

Diggle, P. and Ribeiro, P. J. (2007). *Model-Based Geostatistics.* Springer.

Dillon, M. O., Nakazawa, M., and Leiva, S. G. (2003). The Lomas formations of coastal Peru: Composition and biogeographic history. In Haas, J. and Dillon, M. O., editors, *El Niño in Peru: Biology and Culture over 10,000 Years*, pages 1–9. Field Museum of Natural History, Chicago.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

Eddelbuettel, D. and Balamuta, J. J. (2018). Extending R with C++: A Brief Introduction to Rcpp. *The American Statistician*, 72(1):28–36.

Galletti, C. S., Turner, B. L., and Myint, S. W. (2016). Land changes and their drivers in the cloud forest and coastal zone of Dhofar, Oman, between 1988 and 2013. *Regional Environmental Change*, 16(7):2141–2153.

Garrard, C. (2016). *Geoprocessing with Python.* Manning Publications, Shelter Island, NY. OCLC: ocn915498655.

Gelfand, A. E., Diggle, P., Guttorp, P., and Fuentes, M. (2010). *Handbook of Spatial Statistics.* CRC press.

Gillespie, C. and Lovelace, R. (2016). *Efficient R Programming: A Practical Guide to Smarter Programming.* O'Reilly Media.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*. Applied Geostatistics Series. Oxford University Press, New York.

Graser, A. and Olaya, V. (2015). Processing: A Python Framework for the Seamless Integration of Geoprocessing Tools in QGIS.

Grolemund, G. and Wickham, H. (2016). *R for Data Science*. O'Reilly Media.

Hengl, T. (2007). *A Practical Guide to Geostatistical Mapping of Environmental Variables*. Publications Office, Luxembourg. OCLC: 758643236.

Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B., and Gräler, B. (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ*, 6:e5518.

Hickman, R., Ashiru, O., and Banister, D. (2011). Transitions to low carbon transport futures: Strategic conversations from London and Delhi. *Journal of Transport Geography*, 19(6):1553–1562.

Hijmans, R. J. (2016). *Geosphere: Spherical Trigonometry*.

Hijmans, R. J. (2017). *raster: Geographic Data Analysis and Modeling*. R package version 2.6-7.

Hollander, Y. (2016). *Transport Modelling for a Complete Beginner*. CTthink!

Horni, A., Nagel, K., and Axhausen, K. W. (2016). *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press.

Huang, Z., Chen, Y., Wan, L., and Peng, X. (2017). GeoSpark SQL: An Effective Framework Enabling Spatial Queries on Spark. *ISPRS International Journal of Geo-Information*, 6(9):285.

Huff, D. L. (1963). A Probabilistic Analysis of Shopping Center Trade Areas. *Land Economics*, 39(1):81–90.

Hunziker, P. (2017). *Velox: Fast Raster Manipulation and Extraction*.

Jafari, E., Gemar, M. D., Juri, N. R., and Duthie, J. (2015). Investigation of Centroid Connector Placement for Advanced Traffic Assignment Models with Added Network Detail. *Transportation Research Record: Journal of the Transportation Research Board*, 2498:19–26.

James, G., Witten, D., Hastie, T., and Tibshirani, R., editors (2013). *An Introduction to Statistical Learning: With Applications in R*. Number 103 in Springer Texts in Statistics. Springer, New York. OCLC: ocn828488009.

Jenny, B., Šavrič, B., Arnold, N. D., Marston, B. E., and Preppernau, C. A. (2017). A guide to selecting map projections for world and hemisphere maps. In Lapaine, M. and Usery, L., editors, *Choosing a Map Projection*, pages 213–228. Springer.

Jr, P. J. R. and Diggle, P. J. (2016). *geoR: Analysis of Geostatistical Data*.

Kahle, D. and Wickham, H. (2013). Ggmap: Spatial Visualization with ggplot2. *The R Journal*, 5:144–161.

Kaiser, M. and Morin, T. (1993). Algorithms for computing centroids. *Computers & Operations Research*, 20(2):151–165.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). Kernlab - An S4 Package for Kernel Methods in R. *Journal of Statistical Software*, 11(9).

Knuth, D. E. (1974). Computer Programming As an Art. *Commun. ACM*, 17(12):667–673.

Krainski, E., Gómez Rubio, V., Bakka, H., Lenzi, A., Castro-Camilo, D., Simpson, D., Lindgren, F., and Rue, H. (2018). *Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA*.

Krug, R. M., Roura-Pascual, N., and Richardson, D. M. (2010). Clearing of invasive alien plants under different budget scenarios: Using a simulation model to test efficiency. *Biological invasions*, 12(12):4099–4112.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer, New York. OCLC: ocn827083441.

Lamigueiro, O. P. (2018). *Displaying Time Series, Spatial, and Space-Time Data with R*. Chapman and Hall/CRC, Boca Raton, second edition.

Landa, M. (2008). New GUI for GRASS GIS based on wxPython. *Departament of Geodesy and Cartography*, pages 1–17.

Liu, J.-G. and Mason, P. J. (2009). *Essential Image Processing and GIS for Remote Sensing*. Wiley-Blackwell, Chichester, West Sussex, UK, Hoboken, NJ.

Livingstone, D. N. (1992). *The Geographical Tradition: Episodes in the History of a Contested Enterprise*. John Wiley & Sons Ltd, Oxford, UK ; Cambridge, USA.

Longley, P. (2015). *Geographic Information Science & Systems*. Wiley, Hoboken, NJ, fourth edition.

Longley, P. A., Brooks, S. M., McDonnell, R., and MacMillan, B., editors (1998). *Geocomputation: A Primer*. Wiley, Chichester, Eng. ; New York.

Lovelace, R. and Dumont, M. (2016). *Spatial Microsimulation with R*. CRC Press.

Lovelace, R., Goodman, A., Aldred, R., Berkoff, N., Abbas, A., and Woodcock, J. (2017). The Propensity to Cycle Tool: An open source online system for sustainable transport planning. *Journal of Transport and Land Use*, 10(1).

Majure, J. J. and Gebhardt, A. (2016). *Sgeostat: An Object-Oriented Framework for Geostatistical Modeling in S+*.

Maling, D. H. (1992). *Coordinate Systems and Map Projections.* Pergamon Press, Oxford ; New York, second edition.

McCune, B., Grace, J. B., and Urban, D. L. (2002). *Analysis of Ecological Communities.* MjM Software Design, Gleneden Beach, OR, second edition. OCLC: 846056595.

Meulemans, W., Dykes, J., Slingsby, A., Turkay, C., and Wood, J. (2017). Small Multiples with Gaps. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):381–390.

Meyer, H., Reudenbach, C., Hengl, T., Katurji, M., and Nauss, T. (2018). Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. *Environmental Modelling & Software*, 101:1–9.

Miller, H. J. (2004). Tobler's first law and spatial analysis. *Annals of the Association of American Geographers*, 94(2).

Moreno-Monroy, A. I., Lovelace, R., and Ramos, F. R. (2017). Public transport and school location impacts on educational inequalities: Insights from São Paulo. *Journal of Transport Geography.*

Muenchow, J., Bräuning, A., Rodríguez, E. F., and von Wehrden, H. (2013a). Predictive mapping of species richness and plant species' distributions of a Peruvian fog oasis along an altitudinal gradient. *Biotropica*, 45(5):557–566.

Muenchow, J., Brenning, A., and Richter, M. (2012). Geomorphic process rates of landslides along a humidity gradient in the tropical Andes. *Geomorphology*, 139-140:271–284.

Muenchow, J., Dieker, P., Kluge, J., Kessler, M., and von Wehrden, H. (2018). A review of ecological gradient research in the Tropics: Identifying research gaps, future directions, and conservation priorities. *Biodiversity and Conservation*, 27(2):273–285.

Muenchow, J., Hauenstein, S., Bräuning, A., Bäumler, R., Rodríguez, E. F., and von Wehrden, H. (2013b). Soil texture and altitude, respectively, largely determine the floristic gradient of the most diverse fog oasis in the Peruvian desert. *Journal of Tropical Ecology*, 29(05):427–438.

Muenchow, J., Schratz, P., and Brenning, A. (2017). RQGIS: Integrating R with QGIS for statistical geocomputing. *The R Journal*, 9(2):409–428.

Murrell, P. (2016). *R Graphics.* CRC Press, second edition.

Neteler, M. and Mitasova, H. (2008). *Open Source GIS: A GRASS GIS Approach.* Springer, New York, NY, third edition. OCLC: 255568974.

Nolan, D. and Lang, D. T. (2014). *XML and Web Technologies for Data Sciences with R.* Use R! Springer, New York, NY. OCLC: 841520665.

Obe, R. O. and Hsu, L. S. (2015). *PostGIS in Action*. Manning, Shelter Island, NY, second edition. OCLC: ocn872985108.

Office for National Statistics (2014). Workplace Zones: A new geography for workplace statistics - Datasets. https://data.gov.uk/dataset/workplace-zones-a-new-geography-for-workplace-statistics3.

Openshaw, S. and Abrahart, R. J., editors (2000). *Geocomputation*. CRC Press, London ; New York.

O'Rourke, J. (1998). *Computational Geometry in C*. Cambridge University Press, Cambridge, UK, ; New York, NY, USA, second edition.

Padgham, M., Rudis, B., Lovelace, R., and Salmon, M. (2018). *osmdata: Import 'OpenStreetMap' Data as Simple Features or Spatial Objects*. R package version 0.0.7.

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *The R Journal*.

Pebesma, E. and Bivand, R. (2018). *sp: Classes and Methods for Spatial Data*. R package version 1.3-1.

Pebesma, E. and Graeler, B. (2018). *gstat: Spatial and Spatio-Temporal Geostatistical Modelling, Prediction and Simulation*. R package version 1.1-6.

Pebesma, E., Mailund, T., and Hiebert, J. (2016). Measurement Units in R. *The R Journal*, 8(2):486–494.

Pebesma, E., Nüst, D., and Bivand, R. (2012). The R software environment in reproducible geoscientific research. *Eos, Transactions American Geophysical Union*, 93(16):163–163.

Pebesma, E. J. and Bivand, R. S. (2005). Classes and methods for spatial data in R. *R news*, 5(2):9–13.

Pezanowski, S., MacEachren, A. M., Savelyev, A., and Robinson, A. C. (2018). SensePlace3: A geovisual framework to analyze place–time–attribute information in social media. *Cartography and Geographic Information Science*, 45(5):420–437.

Probst, P., Wright, M., and Boulesteix, A.-L. (2018). Hyperparameters and Tuning Strategies for Random Forest. *arXiv:1804.03515 [cs, stat]*.

Qiu, F., Zhang, C., and Zhou, Y. (2012). The Development of an Areal Interpolation ArcGIS Extension and a Comparative Study. *GIScience & Remote Sensing*, 49(5):644–663.

Ripley, B. D. (2001). Spatial Statistics in R. *R News*, 1(2):14–15.

Rodrigue, J.-P., Comtois, C., and Slack, B. (2013). *The Geography of Transport Systems*. Routledge, London, New York, third edition.

Rowlingson, B., Baddeley, A., Turner, R., and Diggle, P. (2003). Rasp: A

Package for Spatial Statistics. In Hornik, K., editor, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing.*

Rowlingson, B. and Diggle, P. (2017). *Splancs: Spatial and Space-Time Point Pattern Analysis.*

Rowlingson, B. S. and Diggle, P. J. (1993). Splancs: Spatial point pattern analysis code in S-plus. *Computers & Geosciences*, 19(5):627–655.

Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., and Brenning, A. (2018). Performance evaluation and hyperparameter tuning of statistical and machine-learning models using spatial data. *arXiv:1803.11266 [cs, stat].*

Sherman, G. (2008). *Desktop GIS: Mapping the Planet with Open Source Tools.* Pragmatic Bookshelf.

Talbert, R. J. A. (2014). *Ancient Perspectives: Maps and Their Place in Mesopotamia, Egypt, Greece, and Rome.* University of Chicago Press.

Tallon, A. R. (2007). Bristol. *Cities*, 24(1):74–88.

Tennekes, M. (2018). Tmap: Thematic Maps in R. *Journal of Statistical Software, Articles*, 84(6):1–39.

The Economist (2016). The autonomous car's reality check. *The Economist.*

Thiele, J. (2014). R Marries NetLogo: Introduction to the RNetLogo Package. *Journal of Statistical Software*, 58(2):1–41.

Tobler, W. R. (1979). Smooth Pycnophylactic Interpolation for Geographical Regions. *Journal of the American Statistical Association*, 74(367):519–530.

Tomintz, M. N. M., Clarke, G. P., and Rigby, J. E. J. (2008). The geography of smoking in Leeds: Estimating individual smoking rates and the implications for the location of stop smoking services. *Area*, 40(3):341–353.

Tomlin, C. D. (1990). *Geographic Information Systems and Cartographic Modeling.* Prentice Hall, Englewood Cliffs, N.J.

Venables, W., Smith, D., and Team, R. C. (2017). *An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics.*

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S.* Springer, New York, fourth edition.

Visvalingam, M. and Whyatt, J. D. (1993). Line generalisation by repeated elimination of points. *The Cartographic Journal*, 30(1):46–51.

von Wehrden, H., Hanspach, J., Bruelheide, H., and Wesche, K. (2009). Pluralism and diversity: Trends in the use and application of ordination methods 1990-2007. *Journal of Vegetation Science*, 20(4):695–705.

Wegmann, M., Leutner, B., and Dech, S., editors (2016). *Remote Sensing and GIS for Ecologists: Using Open Source Software.* Data in the Wild. Pelagic Publishing, Exeter. OCLC: 945979372.

Wickham, H. (2014a). *Advanced R*. CRC Press.

Wickham, H. (2014b). Tidy Data. *Journal of Statistical Software*, 59(10).

Wickham, H. (2016). *Ggplot2: Elegant Graphics for Data Analysis*. Springer, New York, NY, second edition.

Wieland, T. (2017). Market Area Analysis for Retail and Service Locations with MCI. *The R Journal*, 9(1):298–323.

Wilkinson, L. and Wills, G. (2005). *The Grammar of Graphics*. Springer Science+ Business Media.

Wise, S. (2001). *GIS Basics*. CRC Press.

Wulf, A. (2015). *The Invention of Nature: Alexander von Humboldt's New World*. Alfred A. Knopf, New York.

Xiao, N. (2016). *GIS Algorithms: Theory and Applications for Geographic Information Science & Technology*. London.

Zuur, A., Ieno, E. N., Walker, N., Saveliev, A. A., and Smith, G. M. (2009). *Mixed Effects Models and Extensions in Ecology with R*. Statistics for Biology and Health. Springer-Verlag, New York.

Zuur, A. F., Ieno, E. N., Saveliev, A. A., and Zuur, A. F. (2017). *Beginner's Guide to Spatial, Temporal and Spatial-Temporal Ecological Data Analysis with R-INLA*, volume 1. Highland Statistics Ltd, Newburgh, United Kingdom. OCLC: 993615802.