

DISPLAYING TIME SERIES, SPATIAL, AND SPACE-TIME DATA WITH R

OSCAR PERPIÑÁN LAMIGUEIRO

Contents

Contents	i
1 Introduction	1
1.1 What This Book Is About	1
1.2 What You Will <i>Not</i> Find in This Book	2
1.3 How to Read This Book	3
1.4 R Graphics	4
1.5 ggplot2	6
1.6 Packages	7
1.7 Software Used to Write This Book	8
1.8 About the Author	9
1.9 Acknowledgments	9
 I Time Series	 11
2 Displaying Time Series: Introduction	13
2.1 Packages	14
2.2 Further Reading	16
3 Time on the Horizontal Axis	17
4 Time as a Conditioning or Grouping Variable	19
5 Time as a Complementary Variable	21

6	About the Data	23
II	Spatial Data	25
7	Displaying Spatial Data: Introduction	27
7.1	Packages	28
7.2	Further Reading	33
8	Thematic Maps	35
9	Reference and Physical Maps	37
10	About the Data	39
III	Space-Time Data	41
11	Displaying Spatiotemporal Data: Introduction	43
11.1	Packages	44
11.2	Further Reading	46
12	Spatiotemporal Raster Data	47
13	Spatiotemporal Point Observations	49
	Bibliography	51

Chapter 1

Introduction

1.1 What This Book Is About

A data graphic is not only a static image but also tells a story about the data. It activates cognitive processes that are able to detect patterns and discover information not readily available with the raw data. This is particularly true for time series, spatial, and space-time datasets.

There are several excellent books about data graphics and visual perception theory, with guidelines and advice for displaying information, including visual examples. Let's mention *The Elements of Graphical Data* (Cleveland 1994) and *Visualizing Data* (Cleveland 1993) by W. S. Cleveland, *Envisioning Information* (Tufte 1990) and *The Visual Display of Quantitative Information* (Tufte 2001) by E. Tufte, *The Functional Art* by A. Cairo (Cairo 2012), and *Visual Thinking for Design* by C. Ware (Ware 2008). Ordinarily, they do not include the code or software tools to produce those graphics.

On the other hand, there is a collection of books that provides code and detailed information about the graphical tools available with R. Commonly they do not use real data in the examples and do not provide advice for improving graphics according to visualization theory. Three books are the unquestioned representatives of this group: *R Graphics* by P. Murrell (Murrell 2011), *Lattice: Multivariate Data Visualization with R* by D. Sarkar (Sarkar 2008), and *ggplot2: Elegant Graphics for Data Analysis* by H. Wickham (Wickham 2009).

This book proposes methods to display time series, spatial, and space-time data using R, and aims to be a synthesis of both groups providing code and detailed information to produce high-quality graphics with practical examples.

1.2 What You Will *Not* Find in This Book

- **This is not a book to learn R.**

Readers should have a fair knowledge of programming with R to understand the book. In addition, previous experience with the *zoo*, *sp*, *raster*, *lattice*, *ggplot2*, and *grid* packages is helpful.

If you need to improve your R skills, consider these information sources:

- Introduction to R¹.
- Official manuals².
- Contributed documents³.
- Mailing lists⁴.
- R-bloggers⁵.
- Books related to R⁶ and particularly *Software for Data Analysis* by John M. Chambers (Chambers 2008).

- **This book does not provide an exhaustive collection of visualization methods.**

Instead, it illustrates what I found to be the most useful and effective methods. Notwithstanding, each part includes a section titled “Further Reading” with bibliographic proposals for additional information.

- **This book does not include a complete review or discussion of R packages.**

¹<http://cran.r-project.org/doc/manuals/R-intro.html>

²<http://cran.r-project.org/manuals.html>

³<http://cran.r-project.org/other-docs.html>

⁴<http://www.r-project.org/mail.html>

⁵<http://www.r-bloggers.com>

⁶<http://www.r-project.org/doc/bib/R-books.html>

Their most useful functions, classes, and methods regarding data and graphics are outlined in the introductory chapter of each part, and conveniently illustrated with the help of examples. However, if you need detailed information about a certain aspect of a package, you should read the correspondent package manual or vignette. Moreover, if you want to know additional alternatives, you can navigate through the CRAN Task Views about Time Series⁷, Spatial Data⁸, Spatiotemporal Data⁹, and Graphics¹⁰.

- **Finally, this book is not a handbook of data analysis, geostatistics, point pattern analysis, or time series theory.**

Instead, this book is focused on the exploration of data with visual methods, so it may be framed in the Exploratory Data Analysis approach. Therefore, this book may be a useful complement for superb bibliographic references where you will find plenty of information about those subjects. For example, (Chatfield 2003), (Cressie and Wikle 2011), (Slocum 2005) and (R. S. Bivand, E. J. Pebesma, and Gomez-Rubio 2008).

1.3 How to Read This Book

This book is organized into three parts, each devoted to different types of data. Each part comprises several chapters according to the various visualization methods or data characteristics. The chapters are structured as independent units so readers can jump directly to a certain chapter according to their needs. Of course, there are several dependencies and redundancies between the sets of chapters that have been conveniently signaled with cross-references.

The content of each chapter illustrates how to display a dataset starting with an easy and direct approach. Often this first result is not entirely satisfactory so additional improvements are progressively added. Each step involves additional complexity which, in some cases, can be overwhelming during a first reading. Thus, some sections, marked with the sign ☹, can be safely skipped for later reading.

Although I have done my best to help readers understand the methods and code, you should not expect to understand it after one reading. The

⁷<http://cran.r-project.org/web/views/TimeSeries.html>

⁸<http://cran.r-project.org/web/views/Spatial.html>

⁹<http://cran.r-project.org/web/views/SpatioTemporal.html>

¹⁰<http://cran.r-project.org/web/views/Graphics.html>

key is practical experience, and the best way is to try out the code with the provided data **and** modify it to suit your needs with your own data. There is a website and a code repository to help you in this task.

1.3.1 Website and Code Repository

The book website with the main graphics of this book is located at

<http://oscarperpinan.github.com/spacetime-vis/>

The full code is freely available from the repository:

<https://github.com/oscarperpinan/spacetime-vis>

On the other hand, the datasets used in the examples are either available at the repository or can be freely obtained from other websites. It must be underlined that the combination of code and data freely available allows this book to be fully reproducible.

I have chosen the datasets according to two main criteria:

- They are freely available without restrictions for public use.
- They cover different scientific and professional fields (meteorology and climate research, economy and social sciences, energy and engineering, environmental research, epidemiology, etc.).

The repository and the website can be downloaded as compressed files¹¹, and if you use git, you can clone the repository with

```
git clone https://github.com/oscarperpinan/spacetime-vis.git
```

1.4 R Graphics

There are two distinct graphics systems built into R, referred to as traditional and grid graphics. Grid graphics are produced with the grid package (Murrell 2011), a flexible low-level graphics toolbox. Compared with the traditional graphics model, it provides more flexibility to modify or

¹¹Repository: <https://github.com/oscarperpinan/spacetime-vis/archive/master.zip>, Website: <https://github.com/oscarperpinan/spacetime-vis/archive/gh-pages.zip>

add content to an existent graphical output, better support for combining different outputs easily, and more possibilities for interaction. All the graphics in this book have been produced with the grid graphics model.

Other packages are constructed over it to provide high-level functions, most notably the `lattice` and `ggplot2` packages.

1.4.1 `lattice`

The `lattice` package (Sarkar 2008) is an independent implementation of Trellis graphics, which were mostly influenced by *The Elements of Graphing Data* (Cleveland 1994). Trellis graphics often consist of a rectangular array of panels. The `lattice` package uses a *formula* interface to define the structure of the array of panels with the specification of the variables involved in the plot. The result of a `lattice` high-level function is a `trellis` object.

For bivariate graphics, the formula is generally of the form $y \sim x$ representing a single panel plot with y versus x . This formula can also involve expressions. The main function for bivariate graphics is `xypplot`.

Optionally, the formula may be $y \sim x \mid g1 * g2$ and y is represented against x conditional on the variables $g1$ and $g2$. Each unique combination of the levels of these conditioning variables determines a subset of the variables x and y . Each subset provides the data for a single panel in the Trellis display, an array of panels laid out in columns, rows, and pages.

For example, in the following code, the variable `wt` of the dataset `mtcars` is represented against the `mpg`, with a panel for each level of the categorical variable `am`. The points are grouped by the values of the `cyl` variable.

```
xypplot(wt ~ mpg | am, data = mtcars, groups = cyl)
```

For trivariate graphics, the formula is of the form $z \sim x * y$, where z is a numeric response, and x and y are numeric values evaluated on a rectangular grid. Once again, the formula may include conditioning variables, for example $z \sim x * y \mid g1 * g2$. The main function for these graphics is `levelplot`.

The plotting of each panel is performed by the `panel` function, specified in a high-level function call as the `panel` argument. Each high-level `lattice` function has a default panel function, although the user can create new Trellis displays with custom panel functions.

`lattice` is a member of the recommended packages list so it is commonly distributed with R itself. There are more than 250 packages depending on it, and the most important packages for our purposes (`zoo`, `sp`, and `raster`) define methods to display their classes using `lattice`.

On the other hand, the `latticeExtra` package (Sarkar and Andrews 2012) provides additional flexibility for the somewhat rigid structure of the Trellis framework implemented in `lattice`. This package complements the `lattice` with the implementation of layers via the `layer` function, and superposition of `trellis` objects and layers with the `+.trellis` function. Using both packages, you can define a graphic with the formula interface (under the `lattice` model) and overlay additional content as layers (following the `ggplot2` model).

1.5 ggplot2

The `ggplot2` package (Wickham 2009) is an implementation of the system proposed in *The Grammar of Graphics* (Wilkinson 1999), a general scheme for data visualization that breaks up graphs into semantic components such as scales and layers. Under this framework, the definition of the graphic with `ggplot2` is done with a combination of several functions that provides the components, instead of the formula interface of `=lattice`.

With `ggplot2`, a graphic is composed of:

- A dataset, `data`, and a set of mappings from variables to aesthetics, `aes`.
- One or more layers, each composed of: a geometric object, `geom*`, to control the type of plot you create (points, lines, etc.); a statistical transformation, `stat*`; and a position adjustment (and optionally, additional dataset and aesthetic mappings).
- A scale, `scale*`, to control the mapping from data to aesthetic attributes. Scales are common across layers to ensure a consistent mapping from data to aesthetics.
- A coordinate system, `coords*`.
- Optionally, a faceting specification, `facet*`, the equivalent of Trellis graphics with panels.

The function `ggplot` is typically used to construct a plot incrementally, using the `+` operator to add layers to the existing `ggplot` object. For instance, the following code (equivalent to the previous `lattice` example) uses `mtcars` as the dataset, and maps the `mpg` variable on the x-axis and

the `wt` variable on the y-axis. The geometric object is the point using the `cyl` variable to control the color. Finally, the levels of the `am` variable define the panels of the graphic.

```
ggplot(mtcars, aes(mpg, wt)) +  
  geom_point(aes(colour=factor(cyl))) +  
  facet_grid(. ~ am)
```

This package is increasingly popular, with a list of more than ninety packages depending on it. On the other hand, few packages provide method definitions based on `ggplot2` to display their classes. In our context, only the `zoo` package defines the `autoplot` function based on it.

1.5.1 Comparison between `lattice` and `ggplot2`

Which package to choose is, for a wide range of datasets, a question of personal preferences. You may be interested in a comparison between them published in a series of blog posts¹². However, the major drawback of `ggplot2` is its considerably slower speed when dealing with large datasets¹³, so you should be cautious with large spatial and spatiotemporal data. Consequently, most of the code in Part ?? contains alternatives defined both with `lattice` and with `ggplot2`. However, because of the speed problem and the absence of `ggplot2` functions in the corresponding packages, only a minor fraction of the code in Parts ?? and ?? contains graphics defined with `ggplot2`.

1.6 Packages

Throughout the book, several R packages are used. All of them are available from CRAN, and you must install them before using the code. Most of them are loaded at the start of the code of each chapter, although some of them are loaded later if they are used only inside optional sections (marked with ☺). You should install the last version available at CRAN to ensure correct functioning of the code.

Although the introductory chapter of each part includes a section with an outline of the most relevant packages, some of them deserve to be highlighted here:

¹²<http://learnr.wordpress.com/2009/06/28/ggplot2-version-of-figures-in-lattice-multivari>

¹³Take a look at the time comparison published as the final result of the previous series of blog posts, <http://learnr.files.wordpress.com/2009/08/latbook.pdf>

- `zoo` (Zeileis and Grothendieck 2005) provides infrastructure for time series using arbitrary classes for the time stamps (Section 2.1.1).
- `sp` (E. Pebesma 2012) provides a coherent set of classes and methods for the major spatial data types: points, lines, polygons, and grids (Section 7.1.1). `spacetime` (E. Pebesma 2012) defines classes and methods for spatiotemporal data, and methods for plotting data as map sequences or multiple time series (Section 11.1.1).
- `raster` (R. J. Hijmans 2013) is a major extension of gridded spatial data classes. It provides a unified access method to different raster formats, permitting large objects to be analyzed with the definition of basic and high-level processing functions (Sections 7.1.2 and 11.1.2). `rasterVis` (Perpiñán and R. Hijmans 2013) provides enhanced visualization of raster data with methods for spatiotemporal rasters (Sections 7.1.3 and 11.1.3).
- `gridSVG` (Murrell and Potter 2013) converts any grid scene to an SVG document. The `grid.hyperlink` function allows a hyperlink to be associated with any component of the scene, the `grid.animate` function can be used to animate any component of a scene, and the `grid.garnish` function can be used to add SVG attributes to the components of a scene. By setting event handler attributes on a component, plus possibly using the `grid.script` function to add JavaScript to the scene, it is possible to make the component respond to user input such as mouse clicks.

1.7 Software Used to Write This Book

This book has been written using different computers running Debian GNU Linux and using several gems of open-source software:

- `org-mode` for authoring text and code (Schulte et al. 2012).
- R (R Development Core Team 2013) with Emacs Speaks Statistics (Rossini et al. 2004).
- GNU Emacs as development environment.

1.8 About the Author

During the past 15 years, my main area of expertise has been photovoltaic solar energy systems, with a special interest in solar radiation. Initially I worked as an engineer for a private company and I was involved in several commercial and research projects. The project teams were partly integrated by people with low technical skills who relied on the input from engineers to complete their work. I learned how a good visualization output eased the communication process.

Now I work as a professor and researcher at the university. Data visualization is one of the most important tools I have available. It helps me embrace and share the steps, methods, and results of my research. With students, it is an inestimable partner in helping them understand complex concepts.

I have been using R to simulate the performance of photovoltaic energy systems and to analyze solar radiation data, both as time series and spatial data. As a result, I have developed packages that include several graphical methods to deal with multivariate time series (namely, `solaR` (Perpiñán 2012)=) and space-time data (`rasterVis`).

1.9 Acknowledgments

Writing a book is often described as a solitary activity. It is certainly difficult to write when you are with friends or spending time with your family... although with three little children at home I have learned to write prose and code while my baby wants to learn typing and my daughters need help to share a family of dinosaurs.

Seriously speaking, solitude is the best partner of a writer. But when I am writing or coding I feel I am immersed in a huge collaborative network of past and present contributors. Piotr Kropotkin described it with the following words (Kropotkin 1906):

Thousands of writers, of poets, of scholars, have laboured to increase knowledge, to dissipate error, and to create that atmosphere of scientific thought, without which the marvels of our century could never have appeared. And these thousands of philosophers, of poets, of scholars, of inventors, have themselves been supported by the labour of past centuries. They have been upheld and nourished through life, both physically and mentally, by legions of workers and craftsmen of all sorts.

And Lewis Mumford claimed (Mumford 1934):

Socialize Creation! What we need is the realization that the creative life, in all its manifestations, is necessarily a social product.

I want to express my deepest gratitude and respect to all those women and men who have contributed and contribute to strengthening the communities of free software, open data, and open science. My special thanks go to the people of the R community: users, members of the R Core Development Team, and package developers.

With regard to this book in particular, I would like to thank John Kimmel for his constant support, guidance, and patience.

Last, and most importantly, thanks to Candela, Marina, and Javi, my crazy little shorties, my permanent source of happiness, imagination, and love. Thanks to María, *mi amor, mi cómplice y todo*.

Part I

Time Series

Chapter 2

Displaying Time Series: Introduction

A time series is a sequence of observations registered at consecutive time instants. When these time instants are evenly spaced, the distance between them is called the sampling interval. The visualization of time series is intended to reveal changes of one or more quantitative variables through time, and to display the relationships between the variables and their evolution through time.

The standard time series graph displays the time along the horizontal axis. Several variants of this approach can be found in Chapter 3. On the other hand, time can be conceived as a grouping or conditioning variable (Chapter 4). This solution allows several variables to be displayed together with a scatterplot, using different panels for subsets of the data (time as a conditioning variable) or using different attributes for groups of the data (time as a grouping variable). Moreover, time can be used as a complementary variable that adds information to a graph where several variables are confronted (Chapter 5).

These chapters provide a variety of examples to illustrate a set of useful techniques. These examples make use of several datasets (available at the book website) described in Chapter 6.

2.1 Packages

The CRAN Tasks View “Time Series Analysis”¹ summarizes the packages for reading, visualizing, and analyzing time series. This section provides a brief introduction to the `zoo` and `xts` packages. Most of the information has been extracted from their vignettes, webpages, and help pages. You should read them for detailed information.

Both packages extensively use the time classes defined in R. The interested reader will find an overview of the different time classes in R in (Ripley and Hornik 2001) and (Grothendieck and Petzoldt 2004).

2.1.1 `zoo`

The `zoo` package (Zeileis and Grothendieck 2005) provides an S3 class with methods for indexed totally ordered observations. Its key design goals are independence of a particular index class and consistency with base R and the `ts` class for regular time series.

Objects of class `zoo` are created by the function `zoo` from a numeric vector, matrix, or a factor that is totally ordered by some index vector. This index is usually a measure of time but every other numeric, character, or even more abstract vector that provides a total ordering of the observations is also suitable. It must be noted that this package defines two new index classes, `yearmon` and `yearqtr`, for representing monthly and quarterly data, respectively.

The package defines several methods associated with standard generic functions such as `print`, `summary`, `str`, `head`, `tail`, and `[]` (subsetting). In addition, standard mathematical operations can be performed with `zoo` objects, although only for the intersection of the indexes of the objects.

On the other hand, the data stored in `zoo` objects can be extracted with `coredata`, which drops the index information, and can be replaced by `coredata<-`. The index can be extracted with `index` or `time`, and can be modified by `index<-`. Finally, the `window` and `window<-` methods extract or replace time windows of `zoo` objects.

Two `zoo` objects can be merged by common indexes with `merge` and `cbind`. The `merge` method combines the columns of several objects along the union or the intersection of the indexes. The `rbind` method combines the indexes (rows) of the objects.

¹<http://CRAN.R-project.org/view=TimeSeries>

The `aggregate` method splits a `zoo` object into subsets along a coarser index grid, computes a function (sum is the default) for each subset, and returns the aggregated `zoo` object.

This package provides four methods for dealing with missing observations:

1. `na.omit` removes incomplete observations.
2. `na.contiguous` extracts the longest consecutive stretch of non-missing values.
3. `na.approx` replaces missing values by linear interpolation.
4. `na.locf` replaces missing observations by the most recent non-NA prior to it.

The package defines interfaces to `read.table` and `write.table` for reading, `read.zoo`, and writing, `write.zoo`, `zoo` series from or to text files. The `read.zoo` function expects either a text file or connection as input or a `data.frame`. `write.zoo` first coerces its argument to a `data.frame`, adds a column with the index, and then calls `write.table`.

2.1.2 xts

The `xts` package (Ryan and Ulrich 2013) extends the `zoo` class definition to provide a general time-series object. The index of an `xts` object must be of a time or date class: `Date`, `POSIXct`, `chron`, `yearmon`, `yearqtr`, or `timeDate`. With this restriction, the subset operator `[]` is able to extract data using the ISO:8601² time format notation `CCYY-MM-DD HH:MM:SS`. It is also possible to extract a range of times with a `from/to` notation, where both `from` and `to` are optional. If either side is missing, it is interpreted as a request to retrieve data from the beginning, or through the end of the data object.

Furthermore, this package provides several time-based tools:

- `endpoints` identifies the endpoints with respect to time.
- `to.period` changes the periodicity to a coarser time index.
- The functions `period.*` and `apply.*` evaluate a function over a set of non-overlapping time periods.

²http://en.wikipedia.org/wiki/ISO_8601

2.2 Further Reading

- (Wills 2011) provides a systematic analysis of the visualization of time series, and a section of (Jeffrey Heer, Bostock, and Ogievetsky 2010) summarizes the main techniques to display time series.
- (Cleveland 1994) includes a section about time series visualization with a detailed discussion of the banking to 45° technique and the cut-and-stack method. (J. Heer and Agrawala 2006) propose the multi-scale banking, a technique to identify trends at various frequency scales.
- (Few 2008; J. Heer, Kong, and Agrawala 2009) explain in detail the foundations of the horizon graph (Section 3).
- The *small multiples* concept (Sections ?? and ??) is illustrated in (Tufte 2001; Tufte 1990).
- Stacked graphs are analyzed in (Byron and Wattenberg 2008), and the ThemeRiver technique is explained in (Havre et al. 2002).
- (Cleveland 1994; Friendly and Denis 2005) study the scatterplot matrices (Section ??), and (Carr et al. 1987) provide information about hexagonal binning.
- (Harrower and Fabrikant 2008) discuss the use of animation for the visualization of data. (Few 2007) exposes a software tool resembling the Trendalyzer.
- The D3 gallery ³ shows several great examples of time-series visualizations using the JavaScript library D3.js.

³<https://github.com/mbostock/d3/wiki/Gallery>

Chapter 3

Time on the Horizontal Axis

The most frequent visualization method of a time series uses the horizontal axis to depict the time index. This chapter illustrates several variants to display multivariate time series: multiple time series with different scales, variables with the same scale, and stacked graphs.

Chapter 4

Time as a Conditioning or Grouping Variable

In Section ?? we learned to display the time evolution of multiple time series with different scales. But, what if instead of displaying the time evolution we want to confront the variables between them? Section ?? proposes the scatterplot matrix solution with time as a grouping variable. Section ?? uses an enhanced scatterplot with time as a conditioning variable. Section ?? includes a digression about the hexagonal binning for large datasets.

Chapter 5

Time as a Complementary Variable

Gapminder ¹ is an independent foundation based in Stockholm, Sweden. Its mission is “to debunk devastating myths about the world by offering free access to a fact-based world view.” They provide free online tools, data, and videos “to better understand the changing world.” The initial development of Gapminder was the Trendalyzer software, used by Hans Rosling in several sequences of his documentary “The Joy of Stats.”

The information visualization technique used by Trendalyzer is an interactive bubble chart. By default it shows five variables: two numeric variables on the vertical and horizontal axes, bubble size and color, and a time variable that may be manipulated with a slider. The software uses brushing and linking techniques for displaying the numeric value of a highlighted country.

This software was acquired by Google in 2007, and is now available as a Motion Chart gadget and as the Public Data Explorer.

In this chapter, time will be used as a complementary variable which adds information to a graph where several variables are confronted. We will illustrate this approach with the evolution of the relationship between Gross National Income (GNI) and carbon dioxide (CO₂) emissions for a set

¹<http://www.gapminder.org/>

5 TIME AS A COMPLEMENTARY VARIABLE

of countries extracted from the database of the World Bank Open Data. We will try several solutions to display the relationship between CO₂ emissions and GNI over the years using time as a complementary variable. The final method will produce an animated plot resembling the Trendalyzer solution.

Chapter 6

About the Data

Part II

Spatial Data

Chapter 7

Displaying Spatial Data: Introduction

Spatial data (also known as geospatial data) are directly or indirectly referenced to a location on the surface of the Earth. Their spatial reference is composed of coordinate values and a system of reference for these coordinates. Spatial data are often accessed, manipulated, or analyzed through Geographic Information Systems (GIS).

Real objects represented by GIS data can be divided into two abstractions: discrete objects (e.g., a road or a river) represented with vector data (points, lines, and polygons), and continuous fields (such as elevation or solar radiation) represented with raster data. The `sp` package is the preferred option to use vector data in R, and the `raster` package is the choice for raster data ¹.

This part exposes several examples where vector and raster data are displayed to show geographic location of features and physical landscape features of a place (reference and physical maps, Chapter 9) or a specific variable in the context of a geographic reference (thematic maps, Chapter 8). These examples make use of several datasets (available at the book website) described in Chapter 10.

¹ Although `sp` and `raster` are the most important packages, there are an increasing number of packages designed to work with spatial data. They are summarized in the corresponding CRAN Task View. Read Section 7.2 for details.

7.1 Packages

The CRAN Tasks View “Analysis of Spatial Data”² summarizes the packages for reading, visualizing, and analyzing spatial data. This section provides a brief introduction to `sp`, `raster`, `rasterVis`, `maptools`, `rgdal`, `gstat`, and `maps`. Most of the information has been extracted from their vignettes, webpages, and help pages. You should read them for detailed information.

7.1.1 `sp`

The `sp` package (E. J. Pebesma and R. S. Bivand 2005) provides classes and methods for dealing with spatial data in R. The spatial data classes implemented are points (`SpatialPoints`), grids (`SpatialPixels` and `SpatialGrid`), lines (`Line`, `Lines` and `SpatialLines`), rings, and polygons (`Polygon`, `Polygons`, and `SpatialPolygons`), each of them without data or with data (for example, `SpatialPointsDataFrame` or `SpatialLinesDataFrame`).

Selecting, retrieving, or replacing certain attributes in spatial objects with data is done using standard methods:

- `[` selects rows (items) and columns in the `data.frame`.
- `[[` selects a column from the `data.frame`
- `[[<-` assigns or replaces values to a column in the `data.frame`.

A number of spatial methods are available for the classes in `sp`:

- `coordinates(object) <- value` sets spatial coordinates to create spatial data. It promotes a `data.frame` into a `SpatialPointsDataFrame`. *value* may be specified by a formula, a character vector, or a numeric matrix or `data.frame` with the actual coordinates.
- `coordinates(object, ...)` returns a matrix with the spatial coordinates. If used with `SpatialPolygons` it returns a matrix with the centroids of the polygons.
- `bbox` returns a matrix with the coordinates bounding box.
- `proj4string(object)` and `proj4string(object) <- value` retrieve or set projection attributes on spatial classes.

²<http://CRAN.R-project.org/view=Spatial>

- `spTransform` transforms from one coordinate reference system (geographic projection) to another (requires package `rgdal`).
- `spplot` plots attributes combined with spatial data: Points, lines, grids, polygons.

7.1.2 raster

The `raster` package (R. J. Hijmans 2013) has functions for creating, reading, manipulating, and writing raster data. The package provides general raster data manipulation functions. The package also implements raster algebra and most functions for raster data manipulation that are common in Geographic Information Systems (GIS).

The `raster` package can work with raster datasets stored on disk if they are too large to be loaded into memory. The package can work with large files because the objects it creates from these files only contain information about the structure of the data, such as the number of rows and columns, the spatial extent, and the filename, but it does not attempt to read all the cell values in memory. In computations with these objects, the data are processed in chunks.

The package defines a number of S4 classes. `RasterLayer`, `RasterBrick`, and `RasterStack` are the most important:

- A `RasterLayer` object represents single-layer (variable) raster data. It can be created with the function `raster`. This function is able to create a `RasterLayer` from another object, including another `Raster*` object, or from a `SpatialPixels*` and `SpatialGrid*` object, or even a matrix. In addition, it can create a `RasterLayer` reading data from a file. The `raster` package can use raster files in several formats, some of them via the `rgdal` package. Supported formats for reading include GeoTIFF, ESRI, ENVI, and ERDAS.
- `RasterBrick` and `RasterStack` are classes for multilayer data. A `RasterStack` is a list of `RasterLayer` objects with the same spatial extent and resolution. It can be formed with a collection of files in different locations or even mixed with `RasterLayer` objects that only exist in memory. A `RasterBrick` is truly a multilayered object, and processing it can be more efficient than processing a `RasterStack` representing the same data.

The raster package defines a number of methods for raster algebra with Raster* objects: arithmetic operators, logical operators, and functions such as `abs`, `round`, `ceiling`, `floor`, `trunc`, `sqrt`, `log`, `log10`, `exp`, `cos`, `sin`, `max`, `min`, `range`, `prod`, `sum`, `any`, and `all`. In these functions, Raster* objects can be mixed with numbers.

There are several functions to modify the content or the spatial extent of Raster* objects, or to combine Raster* objects:

- The `crop` function takes a geographic subset of a larger Raster* object. `trim` crops a RasterLayer by removing the outer rows and columns that only contain NA values. `extend` adds new rows and/or columns with NA values.
- The `merge` function merges two or more Raster* objects into a single new object.
- `projectRaster` transforms values of a Raster* object to a new object with a different coordinate reference system.
- With `overlay`, multiple Raster* objects can be combined (for example, multiply them).
- `mask` removes all values from one layer that are NA in another layer, and `cover` combines two layers by taking the values of the first layer except where these are NA.
- `calc` computes a function for a Raster* object. With RasterLayer objects, another RasterLayer is returned. With multilayer objects the result depends on the function: With a summary function (`sum`, `max`, etc.), `calc` returns a RasterLayer object, and a RasterBrick object otherwise.
- `stackApply` computes summary layers for subsets of a RasterStack or RasterBrick.
- `cut` and `reclassify` replace ranges of values with single values.
- `zonal` computes zonal statistics, that is, summarizes a Raster* object using zones (areas with the same integer number) defined by another RasterLayer.

7.1.3 rasterVis

The `rasterVis` package (Perpiñán and R. Hijmans 2013) complements the `raster` package, providing a set of methods for enhanced visualization and interaction. This package defines visualization methods (`levelplot`) for quantitative data and categorical data, both for univariate and multivariate rasters.

It also includes several methods in the frame of the Exploratory Data Analysis approach: scatterplots with `xypLOT`, histograms and density plots with `histogram` and `densityplot`, violin and boxplots with `bwpLOT`, and a matrix of scatterplots with `splom`.

On the other hand, this package is able to display vector fields using arrows, `vectorplot`, or with streamlines (Wegenkittl and Gröller 1997), `streamplot`. In this last method, for each point, *droplet*, of a jittered regular grid, a short streamline portion, *streamlet*, is calculated by integrating the underlying vector field at that point. The main color of each streamlet indicates local vector magnitude (slope). Streamlets are composed of points whose sizes, positions, and color degradation encode the local vector direction (aspect).

7.1.4 maptools

The `maptools` package (R. Bivand and Lewin-Koh 2013) provides a set of tools for manipulating and reading geographic data, in particular ESRI (Environmental Systems Research Institute) shapefiles. The package also provides interface wrappers for exchanging spatial objects with packages such as `PBSmapping`, `spatstat`, `maps`, `RArcInfo`, `Stata tmap`, `WinBUGS`, `Mondrian`, and others. The main functions in the context of this book are

- `readShapePoints` reads data from a points shapefile into a `SpatialPointsDataFrame` object.
- `writePointsShape` writes data from a `SpatialPointsDataFrame` object to a shapefile.
- `readShapeLines` reads data from a line shapefile into a `SpatialLinesDataFrame` object.
- `writeLinesShape` writes data from a `SpatialLinesDataFrame` object to a shapefile.
- `readShapePoly` reads data from a polygon shapefile into a `SpatialPolygonsDataFrame` object.

- `writePolyShape` writes data from a `SpatialPolygonsDataFrame` object to a shapefile.
- `map2SpatialPolygons` and `map2SpatialLines` may be used to convert map objects returned by the `map` function in the `maps` package to the classes defined in the `sp` package.
- `spCbind` provides `cbind`-like methods for `Spatial*DataFrame` and `data.frame` objects.

The topology operations on geometries performed by this package (for example, `unionSpatialPolygons`) use the package `rgeos`, an interface to the Geometry Engine Open Source (GEOS) ³.

7.1.5 `rgdal`

The `rgdal` package (R. Bivand, Keitt, and Rowlingson 2013) provides bindings to the Geospatial Data Abstraction Library (GDAL) ⁴. With `readOGR` and `readGDAL`, both GDAL raster and OGR vector map data can be imported into R, and GDAL raster data and OGR vector data can be exported with `writeGDAL` and `writeOGR`.

In addition, this package provides access to projection and transformation operations from the PROJ.4 library ⁵. This package implements several `spTransform` methods providing transformation between datums and conversion between projections using PROJ.4 projection arguments.

7.1.6 `gstat`

The `gstat` package (E. J. Pebesma 2004) provides functions for geostatistical modeling, prediction, and simulation, including variogram modeling and simple, ordinary, universal, and external drift kriging.

Most of the functionality of this package is beyond the scope of this book. However, some functions must be mentioned:

- `variogram` calculates the sample variogram from data, or for the residuals if a linear model is given. `vgm` generates a variogram and `fit.variogram` fit ranges and/or sills from a variogram model to a sample variogram.

³<http://trac.osgeo.org/geos/>

⁴<http://www.gdal.org/>

⁵<https://trac.osgeo.org/proj/>

- `krige` is the function for simple, ordinary or universal kriging. `gstat` is the function for univariate or multivariate geostatistical prediction.

7.1.7 maps

The `maps` (Becker, Wilks, Brownrigg, and Minka 2013), `mapdata` (Becker, Wilks, and Brownrigg 2013), and `mapproj` (McIlroy et al. 2013) packages are useful to draw or create geographical maps. `mapdata` contains higher resolution databases, and `mapproj` converts latitude/longitude coordinates into projected coordinates.

7.2 Further Reading

- (Slocum 2005) and (Dent, Torguson, and Hodler 2008) are comprehensive books on thematic cartography and geovisualization. They include chapters devoted to data classification, scales, map projections, color theory, typography, and proportional symbol, choropleth, dasymetric, isarithmic, and multivariate mapping. Several resources are available at their accompanying websites⁶.
- (R. S. Bivand, E. J. Pebesma, and Gomez-Rubio 2008) is the essential reference to work with spatial data in R. R. Bivand and E. Pebesma are the authors of the fundamental `sp` package, and they are the authors or maintainers of several important packages such as `gstat`, for geostatistical modeling, prediction, and simulation, `rgdal`, `rgeos` and `maptools`. Chapter 3 is devoted to the visualization of spatial data. Code, figures, and data of the book are available at the accompanying website⁷.
- (Hengl 2009) is an open-access book with seven spatial data analysis exercises. The author is the creator and maintainer of the Spatial-Analyst webpage⁸.
- The CRAN Tasks View “Analysis of Spatial Data”⁹ summarizes the packages for reading, vizualizing, and analyzing spatial data. The

⁶<http://www.pearsonhighered.com/slocum3e/> and <http://highered.mcgraw-hill.com/sites/0072943823/>

⁷<http://www.asdar-book.org/>

⁸<http://spatial-analyst.net>

⁹<http://CRAN.R-project.org/view=Spatial>

packages in development published at R-Forge are listed in the “Spatial Data & Statistics” topic view ¹⁰. The R-SIG-Geo mailing list ¹¹ is a powerful resource for obtaining help.

- The “Spatial Analysis” ¹² and “Kartograph” ¹³ webpages publish a variety of beautiful visualization examples.

¹⁰http://r-forge.r-project.org/softwaremap/trove_list.php?form_cat=353

¹¹<https://stat.ethz.ch/mailman/listinfo/R-SIG-Geo/>

¹²<http://spatialanalysis.co.uk/map-gallery/>

¹³<http://kartograph.org/>

Chapter 8

Thematic Maps

A thematic map focuses on a specific theme or variable, commonly using geographic data such as coastlines, boundaries, and places as points of reference for the variable being mapped. These maps provide specific information about particular locations or areas (proportional symbol mapping and choropleth maps) and information about spatial patterns (isarithmic and raster maps). The following sections illustrate the code you need to produce these maps, with a final section devoted to the visualization of vector fields.

Chapter 9

Reference and Physical Maps

A reference map focuses on the geographic location of features. In these maps, cities are named and major transport routes are identified. In addition, natural features such as rivers and mountains are named, and elevation is shown using a simple color shading. A physical map shows the physical landscape features of a place. Mountains and elevation changes are usually shown with different colors and shades to show relief, using green to show lower elevations and browns for high elevations.

This chapter details how to create a reference map of a northern region of Spain using data from OpenStreetMap and a physical map of Brazil with data from different sources.

Chapter 10

About the Data

Part III

Space-Time Data

Chapter 11

Displaying Spatiotemporal Data: Introduction

Space-time datasets are indexed both in space and in time. The data may consist of a spatial vector object (for example, points or polygons) or raster data at different times. The first case is representative of data from fixed sensors providing measurements abundant in time but sparse in space. The second case is the typical format of satellite imagery, which produces high spatial resolution data sparse in time (E. Pebesma 2012).

There are several visualization approaches of space-time data trying to cope with the four dimensions of the data (Cressie and Wikle 2011).

On the one hand, the data can be conceived as a collection of snapshots at different times. These snapshots can be displayed as a sequence of frames to produce an animation, or can be printed on one page with different panels for each snapshot using the small-multiple technique described repeatedly in previous chapters.

On the other hand, one of the two spatial dimensions can be collapsed through an appropriate statistic (for example, mean or standard deviation) to produce a space-time plot (also known as a Hovmöller diagram). The axes of this graphic are typically longitude or latitude as the x-axis, and time as the y-axis, with the value of the spatial-averaged value of the raster data represented with color.

Finally, the space-time object can be reduced to a multivariate time series (where each location is a variable or column of the time series) and displayed with the time series visualization techniques described in the Part ?? . This approach is directly applicable to space-time data sparse in space (for example, point measurements at different times). However, it is mandatory to use aggregation in the case of raster data. In this case, the multivariate time series is composed of the evolution of the raster data averaged along a certain direction.

The next chapters, focused on raster space-time data (Chapter 12) and point space-time data (Chapter 13), illustrate with examples how to produce animations, multipanel graphics, hovmöller diagrams, and time-series with R.

11.1 Packages

The CRAN Tasks View “Handling and Analyzing Spatiotemporal Data”¹ summarizes the packages for reading, vizualizing, and analyzing space-time data. This section provides a brief introduction to the `spacetime`, `raster`, and `rasterVis` packages. Most of the information has been extracted from their vignettes, webpages, and help pages. You should read them for detailed information.

11.1.1 `spacetime`

The `spacetime` package (E. Pebesma 2012) is built upon the classes and methods for spatial data from the `sp` package , and for time series data from the `xts` package. It defines classes to represent four space-time layouts:

1. STF, STFDF: full space-time grid of observations for spatial features and observation time, with all space-time combinations.
2. STS, STSDF: sparse grid layout, stores only the non-missing space-time combinations on a lattice
3. STI, STIDF: irregular layout, time and space points of measured values have no apparent organisation.
4. STT, STTDF: simple trajectories.

¹<http://cran.r-project.org/web/views/SpatioTemporal.html>

Moreover, `spacetime` provides several methods for the following classes:

- `stConstruct`, `STFDF`, and `STIDF` create objects from single or multiple tables.
- `as` coerces to other spatiotemporal objects, `xts`, `Spatial`, `matrix`, or `data.frame`.
- `[` selects or replaces data values.
- `[` selects spatial or temporal subsets, and data variables.
- `over` retrieves index or data values of one object at the locations and times of another.
- `aggregate` aggregates data values over particular spatial, temporal, or spatiotemporal domains.
- `stplot` creates spatiotemporal plots. It is able to produce multi-panel plots, space-time plots, animations, and time series plots.

11.1.2 raster

The `raster` package (R. J. Hijmans 2013) is able to add time information associated with layers of a `RasterStack` or `RasterBrick` object with the `setZ` function. This information can be extracted with `getZ`.

If a `Raster*` object includes this information, the `zApply` function can be used to apply a function over a time series of layers of the object.

11.1.3 rasterVis

`rasterVis` (Perpiñán and R. Hijmans 2013) provides three methods to display spatiotemporal rasters:

1. `hovmoller` produces Hovmöller diagrams (Hovmöller 1949). The axes of this kind of diagram are typically longitude or latitude (x-axis) and time (ordinate or y-axis) with the value of some aggregated field represented through color. However, the user can define the direction with `dirXY` and the summary function with `FUN`.
2. `horizonplot` creates horizon graphs (Few 2008), with many time series displayed in parallel by cutting the vertical range into segments and overplotting them with color representing the magnitude and direction of deviation. Each time series corresponds to a geographical zone defined with `dirXY` and averaged with `zonal`.

3. `xypLOT` displays conventional time series plots. Each time series corresponds to a geographical zone defined with `dirXY` and aggregated with `zonal`.

On the other hand, the `histogram`, `densityplot`, and `bwplot` methods accept a `FUN` argument to be applied to the `z` slot of `Raster*` object (defined by `setZ`). The result of this function is used as the grouping variable of the plot to create different panels.

11.2 Further Reading

- (Cressie and Wikle 2011) is a systematic approach to key quantitative techniques on statistics for spatiotemporal data. The book begins with separate treatments of temporal data and spatial data, and later combines these concepts to discuss spatiotemporal statistical methods. There is a chapter devoted to exploratory methods, including visualization techniques.
- (E. Pebesma 2012) presents the `spacetime` package, which implements a set of classes for spatiotemporal data. This paper includes examples that illustrate how to import, subset, coerce, and export spatiotemporal data, proposes several visualization methods, and discusses spatiotemporal geostatistical interpolation.
- (Slocum 2005) (previously cited in Chapter 7.2) includes a chapter about map animation, discussing several approaches for displaying spatiotemporal data.
- (Hengl 2009) (previously cited in Chapter 7.2) includes a working example with spatiotemporal data to illustrate space-time variograms and interpolation.
- (Harrower and Fabrikant 2008) explore the role of animation in geographic visualization and outline the challenges, both conceptual and technical, involved in the creation and use of animated maps.
- The CRAN Tasks View “Handling and Analyzing Spatiotemporal Data”² summarizes the packages for reading, visualizing, and analyzing space-time data. The R-SIG-Geo mailing list³ is a powerful resource for obtaining help.

²<http://cran.r-project.org/web/views/SpatioTemporal.html>

³<https://stat.ethz.ch/mailman/listinfo/R-SIG-Geo/>

Chapter 12

Spatiotemporal Raster Data

Chapter 13

Spatiotemporal Point Observations

Bibliography

- Becker, Richard A., Allan R. Wilks, and Ray Brownrigg (2013). *mapdata: Extra Map Databases*. R package version 2.2-2. URL: <http://CRAN.R-project.org/package=mapdata>.
- Becker, Richard A., Allan R. Wilks, Ray Brownrigg, and Thomas P Minka (2013). *maps: Draw Geographical Maps*. R package version 2.3-3. URL: <http://CRAN.R-project.org/package=maps>.
- Bivand, Roger S., Edzer J. Pebesma, and Virgilio Gomez-Rubio (2008). *Applied Spatial Data Analysis with R*. Springer, New York. URL: <http://www.asdar-book.org/>.
- Bivand, Roger, Tim Keitt, and Barry Rowlingson (2013). *rgdal: Bindings for the Geospatial Data Abstraction Library*. R package version 0.8-11. URL: <http://CRAN.R-project.org/package=rgdal>.
- Bivand, Roger and Nicholas Lewin-Koh (2013). *maptools: Tools for Reading and Handling Spatial Objects*. R package version 0.8-26. URL: <http://CRAN.R-project.org/package=maptools>.
- Byron, Lee and Martin Wattenberg (2008). *Stacked Graphs – Geometry & Aesthetics*. Tech. rep. URL: http://www.leebyron.com/else/streamgraph/download.php?file=stackedgraphs_byron_wattenberg.pdf.
- Cairo, Alberto (2012). *The Functional Art: An Introduction to Information Graphics and Visualization*. New Riders Publishing, Aarhus, Denmark.
- Carr, D. B. et al. (1987). "Scatterplot Matrix Techniques for Large N". English. In: *Journal of the American Statistical Association* 82.398, pp. 424–436. ISSN: 01621459. URL: <http://www.jstor.org/stable/2289444>.
- Chambers, John M. (2008). *Software for Data Analysis: Programming with R*. ISBN 978-0-387-75935-7. Springer, New York. URL: <http://stat.stanford.edu/~jmc4/Rbook/>.

- Chatfield, Chris (2003). *The Analysis of Time-series: An Introduction*. CRC Press, Boca Raton, FL., p. 333.
- Cleveland, W. S. (1993). *Visualizing Data*. Summit, NJ: Hobart Press.
- (1994). *The Elements of Graphing Data*. Murray Hill, NJ.: AT&T, Bell Laboratories.
- Cressie, N. and C.K. Wikle (2011). *Statistics for Spatio-Temporal Data*. Wiley Series in Probability and Statistics. Wiley, New York. ISBN: 9780471692744.
- Dent, B., J. Torguson, and T. Hodler (2008). *Cartography: Thematic Map Design*. McGraw-Hill Education, New York. ISBN: 9780072943825.
- Few, S. (2007). *Visualizing Change: An Innovation in Time-Series Analysis*. Tech. rep. Perceptual Edge, Berkeley, CA. URL: http://www.perceptualedge.com/articles/visual_business_intelligence/visualizing_change.pdf.
- (2008). *Time on the Horizon*. Tech. rep. Perceptual Edge, Berkeley, CA. URL: http://www.perceptualedge.com/articles/visual_business_intelligence/time_on_the_horizon.pdf.
- Friendly, Michael and Daniel Denis (2005). “The early origins and development of the scatterplot”. In: *Journal of the History of the Behavioral Sciences* 41.2, pp. 103–130. ISSN: 1520-6696. DOI: 10.1002/jhbs.20078.
- Grothendieck, Gabor and Thomas Petzoldt (2004). “R Help Desk: Date and Time Classes in R”. In: *R News* 4.1, pp. 29–32. URL: http://CRAN.R-project.org/doc/Rnews/Rnews_2004-1.pdf.
- Harrower, M. and S. I. Fabrikant (2008). “The Role of Map Animation in Geographic Visualization”. In: *Geographic Visualization: Concepts, Tools and Applications*. Ed. by M. Dodge, M. McDerby, and Turner M. Chichester, UK: Wiley, pp. 49–65. URL: <http://www.zora.uzh.ch/8979/>.
- Havre, S. et al. (2002). “ThemeRiver: Visualizing Thematic Changes in Large Document Collections”. In: *IEEE Transactions on Visualization and Computer Graphics* 8.1, pp. 9–20. ISSN: 1077-2626. DOI: 10.1109/2945.981848.
- Heer, J. and M. Agrawala (2006). “Multi-Scale Banking to 45 Degrees”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5, pp. 701–708. ISSN: 1077-2626. DOI: 10.1109/TVCG.2006.163. URL: <http://vis.berkeley.edu/papers/banking/2006-Banking-InfoVis.pdf>.
- Heer, J., N. Kong, and M. Agrawala (2009). “Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations”. In: *ACM Human Factors in Computing Systems (CHI)*, pp. 1303–1312. URL: <http://vis.berkeley.edu/papers/horizon/2009-TimeSeries-CHI.pdf>.

- Heer, Jeffrey, Michael Bostock, and Vadim Ogievetsky (2010). "A tour through the visualization zoo". In: *Communications of the ACM* 53.6, pp. 59–67. ISSN: 0001-0782. DOI: [10.1145/1743546.1743567](https://doi.org/10.1145/1743546.1743567). URL: <http://doi.acm.org/10.1145/1743546.1743567>.
- Hengl, T. (2009). *A Practical Guide to Geostatistical Mapping*. University of Amsterdam, Amsterdam. URL: <http://spatial-analyst.net/book/>.
- Hijmans, Robert J. (2013). *raster: Geographic Analysis and Modeling with Raster Data*. R package version 2.1-66. URL: <http://CRAN.R-project.org/package=raster>.
- Hovmöller, Ernest (1949). "The Trough-and-Ridge diagram". In: *Tellus* 1.2, pp. 62–66. ISSN: 2153-3490. DOI: [10.1111/j.2153-3490.1949.tb01260.x](https://doi.org/10.1111/j.2153-3490.1949.tb01260.x).
- Kropotkin, Piotr (1906). *The Conquest of Bread*. G. P. Putnam's Sons. URL: http://en.wikisource.org/wiki/The_Conquest_of_Bread.
- McIlroy, Doug et al. (2013). *mapproj: Map Projections*. R package version 1.2-1. URL: <http://CRAN.R-project.org/package=mapproj>.
- Mumford, L. (1934). *Technics and Civilization*. San Diego, CA.: Harcourt, Brace & Company, Inc.
- Murrell, Paul (2011). *R Graphics*. 2nd. The R Series. Boca Raton, FL.: Chapman & Hall/CRC, p. 546.
- Murrell, Paul and Simon Potter (2013). *gridSVG: Export grid graphics as SVG*. R package version 1.3-1. URL: <http://CRAN.R-project.org/package=gridSVG>.
- Pebesma, Edzer (2012). "spacetime: Spatio-Temporal Data in R". In: *Journal of Statistical Software* 51.7, pp. 1–30. ISSN: 1548-7660. URL: <http://www.jstatsoft.org/v51/i07>.
- Pebesma, Edzer J. (2004). "Multivariable Geostatistics in S: The gstat Package". In: *Computers and Geosciences* 30, pp. 683–691.
- Pebesma, Edzer J. and Roger S. Bivand (2005). "Classes and methods for spatial data in R". In: *R News* 5.2, pp. 9–13. URL: <http://CRAN.R-project.org/doc/Rnews/>.
- Perpiñán, Oscar (2012). "solaR: Solar Radiation and Photovoltaic Systems with R". In: *Journal of Statistical Software* 50.9, pp. 1–32. URL: <http://www.jstatsoft.org/v50/i09/>.
- Perpiñán, Oscar and Robert Hijmans (2013). *rasterVis: Visualization Methods for the raster Package*. R package version 0.24. URL: <http://CRAN.R-project.org/package=rasterVis>.
- R Development Core Team (2013). *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria. URL: <http://www.R-project.org>.

- Ripley, Brian D. and Kurt Hornik (2001). "Date-Time Classes". In: *R News* 1.2, pp. 8–11. URL: http://CRAN.R-project.org/doc/Rnews/Rnews_2001-2.pdf.
- Rossini, A. J. et al. (2004). "Emacs Speaks Statistics: A Multiplatform, Multipackage Development Environment for Statistical Analysis". In: *Journal of Computational and Graphical Statistics* 13.1, pp. 247–261. DOI: 10.1198/1061860042985. eprint: <http://www.tandfonline.com/doi/pdf/10.1198/1061860042985>. URL: <http://ess.r-project.org/>.
- Ryan, Jeffrey A. and Joshua M. Ulrich (2013). *xts: eXtensible Time Series*. R package version 0.9-5. URL: <http://CRAN.R-project.org/package=xts>.
- Sarkar, Deepayan (2008). *Lattice: Multivariate Data Visualization with R*. ISBN 978-0-387-75968-5. New York: Springer. URL: <http://lmdvr.r-forge.r-project.org>.
- Sarkar, Deepayan and Felix Andrews (2012). *latticeExtra: Extra Graphical Utilities Based on lattice*. R package version 0.6-24. URL: <http://CRAN.R-project.org/package=latticeExtra>.
- Schulte, Eric et al. (2012). "A Multi-Language Computing Environment for Literate Programming and Reproducible Research". In: *Journal of Statistical Software* 46.3, pp. 1–24. ISSN: 1548-7660. URL: <http://www.jstatsoft.org/v46/i03>.
- Slocum, Terry A. (2005). *Thematic Cartography and Geographic Visualization*. Englewood Cliffs, NJ.: Prentice Hall. ISBN: 9780130351234.
- Tufte, E. R. (1990). *Envisioning information*. Cheshire, CT.: Graphic Press.
- (2001). *The Visual Display of Quantitative Information*. Cheshire, CT.: Graphic Press.
- Ware, Colin (2008). *Visual Thinking for Design*. Burlington, MA.: Morgan Kaufmann Pub. ISBN: 9780123708960.
- Wegenkittl, Rainer and Eduard Gröller (1997). "Fast Oriented Line Integral Convolution for Vector Field Visualization via the Internet". In: *Proceedings of the 8th Conference on Visualization'97*. IEEE Computer Society Press, pp. 309–316. URL: http://christl.cg.tuwien.ac.at/research/vis/dynsys/frolic/frolic_crc.pdf.
- Wickham, Hadley (2009). *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer. ISBN: 978-0-387-98140-6. URL: <http://ggplot2.org/book/>.
- Wilkinson, Leland (1999). *The Grammar of Graphics*. New York: Springer. ISBN: 0387987746.

- Wills, G. (2011). *Visualizing Time: Designing Graphical Representations for Statistical Data*. Statistics and Computing. New York: Springer. ISBN: 9780387779065.
- Zeileis, Achim and Gabor Grothendieck (2005). "zoo: S3 Infrastructure for Regular and Irregular Time Series". In: *Journal of Statistical Software* 14.6, pp. 1–27. URL: <http://www.jstatsoft.org/v14/i06/>.