Geocodes and Simple Features Data

David W. Gerbing
The School of Business
Portland State University

April 20, 2026

Excerpted from Chapter 8 of the forthcoming 2nd edition:

Data Visualization: Derive Meaning from Data

CRC Press

8.1 Overview

variable-based visualizations: Show how a variable is distributed, summarized, or related to another variable.

structural visualizations: Show how entities occupy space, are connected, or move.

Previous chapters introduced *variable-based visualizations*, which show how a variable is distributed, summarized, or related to another variable. Bar charts, histograms, scatter plots, and time-series plots all share the same underlying framework: data organized in a rectangular table, with each column representing a variable and each row representing an observation.

In this chapter we turn to another class of visualizations. *Structural visualizations* show how entities *occupy space*, are *connected*, or *move*. These visualizations reveal relationships that are not naturally expressed as variables in a rectangular data table. The variable-based visualizations presented previously answer questions such as, What is the distribution of x? Or, How does y change with x? Structural visualizations answer questions such as:

- ▷ Where are data points located geographically, and how does location relate to their values?
- ▷ How do people, products, or funds flow from one category or stage to another?
- ▷ How are entities related or connected within a system?

This chapter presents each of the following structural visualizations in more detail.

Maps. A *map* places data in geographic space, showing where observations occur. Unlike a scatter plot, the x and y coordinates correspond to longitude and latitude and are often drawn on a base map of geographic boundaries. Packages such as `mapview` and `leaflet` provide simple interactive mapping aligned with the `lessR` philosophy: powerful results from minimal input.

Networks. A *network* visualization displays connections among entities such as people, web pages, or organizations. These entities are represented as nodes linked by edges. Even if a dedicated network function is not implemented directly in `lessR`, the concept belongs here because it visualizes *structure rather than variable values*. Network diagrams answer questions about relationships, influence, and connectivity within a system.

Flows. A *flow visualization* shows how quantities move between categories or states. Sankey and alluvial diagrams are common examples, in which the widths of the bands represent the size of the flow from each source to each destination. In `lessR`, the `Flows()` function provides a concise interface for creating interactive flow diagrams with Plotly, extending the same simplicity that `Chart()`, `X()`, and `XY()` bring to variable-based visualizations. Flows differ from time series in an important way: time implies a sequence, whereas a flow represents *movement between categories*, not movement across time.

Together, maps, networks, and flows broaden the framework of data visualization developed throughout this book. The first seven chapters focus on *variable-based* visualizations, which examine relationships among values stored in a data table. This chapter turns to *structural* visualizations, which instead reveal systems of location, connection, and movement. Both perspectives, variable-based and structural, are essential for deriving meaning from data.

8.2 Maps

In this section, we create maps of various geographic entities, including countries, cities, and smaller areas. We can also enhance these maps by adding additional information, such as population density or economic indicators.

8.2.1 Map Basics

Projections

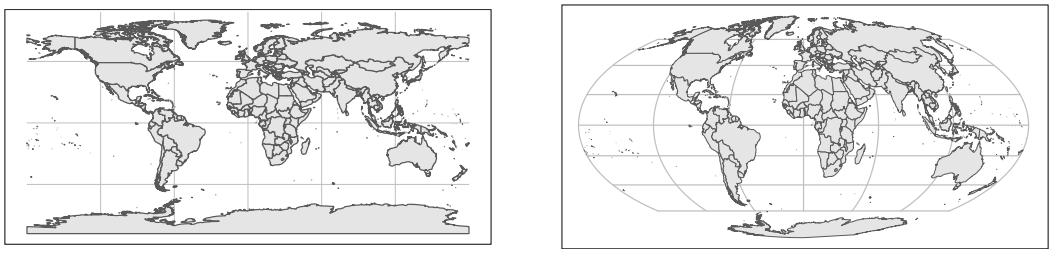
projection: Flatten the coordinates of latitude and longitude that describe a position on the Earth's three-dimensional surface into two dimensions.

The Earth is approximately spherical, but maps are drawn on flat surfaces, usually paper or a computer screen. To create a map, we *project* spherical coordinates, which specify locations on a curved three-dimensional surface, onto the two-dimensional x,y Cartesian coordinates of a flat surface.

All maps, whether of a small region or of the entire Earth, are projections. The difficulty is that a flat surface cannot represent a spherical object without distortion. Of the many competing projections, each has its own limitations. A transformation from a spherical surface to a flat one cannot simultaneously preserve area, direction, distance, and shape. Each projection therefore compromises at least one of these properties while attempting to minimize distortion in the others. The larger the mapped area, the greater the distortion. Consequently, a world map necessarily exhibits more distortion than a city map.

world maps directions for creating, Section 8.2.7, p. 263

Figure 8.1 presents two world maps based on two different projections.



(a) Lat/long Projection **(b)** Mollweide Projection

Figure 8.1: World maps created from the same data but two different projections.

Figure 8.1a presents the simplest possible projection, which maps longitude and latitude directly into the map's rectangular grid `<x,y>` coordinates. This simple rectangular longitude–latitude rendering of the world is more formally known as the Plate Carrée projection, French for "square plan". This projection renders the Earth as a flat rectangle but introduces large distortions, especially away from the equator.

Figure 8.1b shows the Mollweide projection, an equal-area projection that preserves the relative sizes of regions but not angles. As a result, the continents appear in correct proportion by area, with much less distortion of size compared to Plate Carrée, though their shapes and angular relationships are not preserved. The Mollweide

projection applies a nonlinear transformation of latitude and longitude into the map's `<x,y>` plane, producing a more balanced view of the world.

Many projections are also available for online viewing.

> *Interactive world maps from different projections*
>
> *URL*: `https://www.jasondavies.com/maps/transition/`
> *URL*: `http://projections.mgis.psu.edu`

Or, explore further and create custom projections with the free cross-platform downloadable Java software from `flexprojector.com`.

Map types

There are many types of maps. The following are among the most common.

- *Political map* or *country map*: Shows political and administrative boundaries, such as countries, states within countries, and cities within states. When plotted symbols represent geographic entities such as cities, the size of a symbol can also reflect the value of another variable, such as population. For a sufficiently small area that includes streets and highways, this type of map can be called a *street map*.
- *Choropleth map* or *thematic map*: Fills different parts of the map with varying color intensities, where the color intensity reflects the value of another variable.
- *Physical map* or *terrain map*: Shows natural landscape features of the Earth, often using imagery of the world, which today includes satellite images.

Visualization modes

Digital images are stored in one of two primary formats: vector or raster.

Vector image: Composed of paths defined by mathematical formulas, including points, lines, and curves.

Vector images are resolution-independent, so they scale perfectly from small to large with no distortion because they consist of mathematical descriptions of objects. To view a vector image is to view the display of mathematical objects rendered at a given resolution. Vector images are informally referred to as drawings, roughly analogous to a human-created drawing with pen and paper, but with the added advantage of perfect scalability. Common file types include `.pdf` and `.eps`. However, vector images cannot achieve the finest level of detail and gradation that a photograph can provide, which is an example of the other primary type of image.

Raster image: Composed of a grid of pixels, each having a specific color value.

A photograph is a *raster image*, composed of a large number of tiny pixels or dots. The disadvantages are that raster images can require much more storage space than vector images and can distort when scaled to larger sizes. At its native resolution, however, a raster image can achieve fine detail, including subtle textures and gradual changes in color. A raster image is also called a *bit-mapped image* or, more informally, a painting or photograph. Common file types include `.jpeg`, `.jpg`, and `.png`. See Figure 8.2 for an example.

Historically, detailed maps were only created from raster images. However, with the advancement of computer imaging technology, we can now construct high-quality

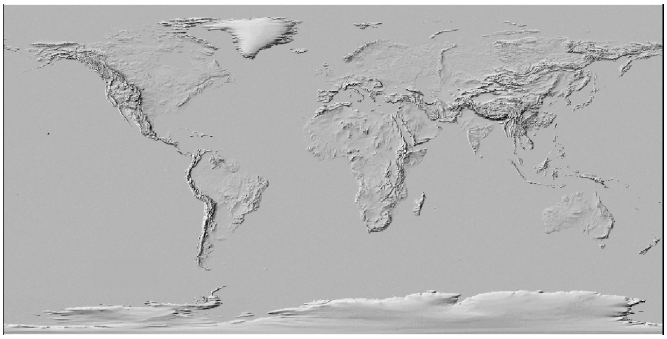


Figure 8.2: Grayscale raster image scale relief map of the world from Natural Earth.

maps from vector images. While raster maps can still be created from visualization systems, the success of high-quality vector images has shifted our focus to the more flexible vector-based mapping.

8.2.2 Spatial data

spatial data: Describes geophysical features and locations, such as the boundaries between countries, altitudes, and shorelines.

Drawing a map requires the following information.

- ▷ The objects to be drawn on the map.
- ▷ The location of each object.

Geocodes

geocode: An object's location expressed in longitude and latitude.

Express the locations of the relevant geographic features in terms of longitude and latitude. The location of each object in these coordinates is its *geocode*. In general, we have two ways to obtain geocodes.

- ▷ Use an online service to geocode specific addresses that we provide, which allows the resulting map to be customized to the locations of interest.
- ▷ Access data files on the web that already include geocodes for existing geographic features, such as cities around the world.

Consider first the option of obtaining customized geocodes. Create a file that contains two columns: the names of the locations to map and their corresponding addresses. A location might be a city, a specific address within a city, a mountain, a lake, or some other relevant geographic feature. These location data are processed as an R data frame, so store them in any file format that R can read.

For example, suppose we create a map of the locations of several universities and colleges in Portland, Oregon. Begin with two columns of data entered into a spreadsheet. The first variable, *school*, is the name of the feature. Its location is specified by the second variable, *address*. Once obtained from a geocoding service, the resulting geocode data file contains the original locations of interest plus two additional variables: the corresponding coordinates that locate the geographic features on the Earth.

Simple features (`sf`) data: Data that uses geometric figures such as points, lines, or polygons and their respective coordinates to describe the features of geographic objects.

Simple features data

To draw a map, we need more than the objects' geocodes. We also need the type of object to be drawn. To represent this spatial data, geographers developed specific data table formats, of which the most commonly encountered is called *simple features* (`sf`). Draw a map by entering a `sf` data frame into a mapping function.

A `sf` data table describes vector data. Each row of a `sf` data table represents a single spatial object and the coordinates that represent that object on the map. For example, draw a point on the map located at its associated coordinates, or draw the polygon that outlines the shape of a country.

The transformation of the original geocoded data frame into a `sf` data frame incorporates the longitude and latitude values into a variable named `geometry`. These data values include the coordinates and the type of geometric object to be represented on the map. To plot the location of a specific school, the type of object drawn for this particular row of data is a `POINT`.

We can build this `sf` data file on our computer step-by-step, or we can directly access the relevant `sf` data file from the internet. For example, when mapping an entire country and its corresponding states or provinces, the most straightforward approach is to download the `sf` file that already contains the needed spatial information. From the `sf` data, we can create our maps.

8.2.3 Political Map: Portland

Custom geocodes

There are multiple sources of custom geocodes. Here, obtain the geocodes with the `geocode()` function from the `tidygeocoder` package. Present the parameter `address` for each object as a single character string or divide it into separate variables according to parameters `street`, `city`, `state`, `postalcode`, and `country`. Name the variables of your choice in your data table, but reference them with these given parameter names when calling the `geocode()` function.

Figure 8.3 shows the information stored in an Excel file from which the geocodes will be obtained. The needed location information is provided as a single address variable. Other variables relevant to mapping can also be included, such as the number of students enrolled at each school.

school	address	students
Portland State University	1825 SW Broadway, Portland, OR 97201	12490
University of Portland	5000 N Willamette Blvd, Portland, OR 97203	3700
Reed College	3203 SE Woodstock Blvd, Portland, OR 97202	1458
Lewis and Clark College	615 S Palatine Hill Rd, OR 97219	3520

Figure 8.3: Locations from which to obtain the corresponding geocodes.

To display the `tibble` as a standard R data frame, enclose the name of the `tibble` with the function `data.frame()`.

Listing 8.1 shows the resulting geocoded data frame from calling the function `geocode()`. The function returns the results in the form of the `tidyverse` version of a data frame, the `tibble` named *pdx*, with four rows of data and five columns.

```
> pdx
# A tibble: 4 × 5
  school                    address     students   lat  long
  <chr>                     <chr>          <int> <dbl> <dbl>
1 Portland State University 1825 SW Br...    12490  45.5 -123.
2 University of Portland    5000 N Wil...     3700  45.6 -123.
3 Reed College              3203 SE Wo...     1458  45.5 -123.
4 Lewis and Clark College   615 S Pala...     3520  45.5 -123.
```

Listing 8.1: `sf tibble` data frame for plotting the location of some university and college in Portland, OR.

Simple features data

coordinate reference system (CRS): Defines how locations on the Earth are represented spatially. A CRS includes a datum, the mathematical model of the Earth used for specifying location, and may also include a projection that transforms geographic coordinates (latitude and longitude) to flat Cartesian coordinates for mapping.

Specifying the location of a geographic object requires adopting a coordinate reference system (CRS), which defines how spatial coordinates are referenced for mapping. A CRS includes a model of the Earth's shape and may also include a projection that transforms the Earth's curved surface to a flat one. CRS definitions and transformations in R rely on the open-source PROJ library (PROJ contributors, 2025).

For many applications, the global CRS EPSG:4326, also referred to as WGS84, is the default CRS. It expresses locations in latitude and longitude and provides the basis for worldwide GPS. However, alternative global and regional CRS definitions are available and may be essential when high local accuracy is required.

Within R, the function `st_as_sf()` from the `sf` package transforms a geocoded data frame into an `sf` data frame by constructing the `geometry` variable that stores the spatial coordinates. Listing 8.2 shows the resulting `tibble` version of the `sf` data frame, or, to display it as a standard data frame, `data.frame(pdx.sf)`.

```
> pdx.sf
Simple feature collection with 4 features and 3 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -122.7266 ymin: 45.45069 xmax: -122.631 ymax: 45.57242
Geodetic CRS:  WGS 84
# A tibble: 4 × 4
  school          address students             geometry
* <chr>           <chr>      <int>          <POINT [°]>
1 Portland State... 1825 S...    12490 (-122.6844 45.51179)
2 University of ... 5000 N...     3700 (-122.7266 45.57242)
3 Reed College    3203 S...     1458   (-122.631 45.4815)
4 Lewis and Clar... 615 S ...     3520 (-122.6713 45.45069)
```

Listing 8.2: Data frame in the form of a `tibble` with geocodes.

Interactive maps

`mapview()` interfaces to the interactive `leaflet` (2025) mapping system, much as `lessR` simplifies work done directly in base R.

The `mapview()` function from the package of the same name provides a straightforward way to create interactive maps in a variety of styles. The simplest call to `mapview()` references only the corresponding `sf` data frame, from which the entire map is constructed. The function automatically scales the map to include the area