Chapter 6

Scales, axes and legends

6.1 Introduction

Scales control the mapping from data to aesthetics. They take your data and turn it into something that you can perceive visually: e.g., size, colour, position or shape. Scales also provide the tools you use to read the plot: the axes and legends (collectively known as guides). Formally, each scale is a function from a region in data space (the domain of the scale) to a region in aesthetic space (the range of the scale). The domain of each scale corresponds to the range of the variable supplied to the scale, and can be continuous or discrete, ordered or unordered. The range consists of the concrete aesthetics that you can perceive and that R can understand: position, colour, shape, size and line type. If you blinked when you read that scales map data both to position and colour, you are not alone. The notion that the same kind of object is used to map data to positions and symbols strikes some people as unintuitive. However, you will see the logic and power of this notion as you read further in the chapter.

The process of scaling takes place in three steps, transformation, training and mapping, and is described in Section 6.2. Without a scale, there is no way to go from the data to aesthetics, so a scale is required for every aesthetic used on the plot. It would be tedious to manually add a scale every time you used a new aesthetic, so whenever a scale is needed ggplot2 will add a default. You can generate many plots without knowing how scales work, but understanding scales and learning how to manipulate them will give you much more control. Default scales and how to override them are described in Section 6.3.

Scales can be roughly divided into four categories: position scales, colour scales, the manual discrete scale and the identity scale. The common options and most important uses are described in Section 6.4. The section focusses on giving you a high-level overview of the options available, rather than expanding on every detail in depth. Details about individual parameters are included in the online documentation.

The other important role of each scale is to produce a guide that allows the viewer to perform the inverse mapping, from aesthetic space to data space,
and read values off the plot. For position aesthetics, the axes are the guides; for all other aesthetics, legends do the job. Unlike other plotting systems, you have little direct control over the axis or legend: there is no `gglegend()` or `ggaxis()` to call to modify legends or axes. Instead, all aspects of the guides are controlled by parameters of the scale. Axes and legends are discussed in Section 6.5.

Section 6.6 concludes the chapter with pointers to other academic work that discusses some of the things you need to keep in mind when assigning variables to aesthetics.

### 6.2 How scales work

To describe how scales work, we will first describe the domain (the data space) and the range (the aesthetic space), and then outline the process by which one is mapped to the other.

Since an input variable is either discrete or continuous, the domain is either a set of values (stored as a factor, character vector or logical vector) or an interval on the real line (stored as a numeric vector of length 2). For example, in the mammals sleep dataset (`msleep`), the domain of the discrete variable `vore` is \{carni, herbi, omni\}, and the domain of the continuous variable `bodywt` is \([0.005, 6.654]\). We often think of these as data ranges, but here we are focussing on their nature as input to the scale, i.e., as a domain of a function.

The range can also be discrete or continuous. For discrete scales, it is a vector of aesthetic values corresponding to the input values. For continuous scales, it is a 1d path through some more complicated space. For example, a colour gradient interpolates linearly from one colour to another. The range is either specified by the user when the scale is created, or by the scale itself.

The process of mapping the domain to the range includes the following stages:

- **transformation**: (for continuous domain only). It is often useful to display a transformation of the data, such as a logarithm or square root. Transformations are described in more depth in Section 6.4.2. After any transformations have been applied, the statistical summaries for each layer are computed based on the transformed data. This ensures that a plot of \(\log(x)\) vs. \(\log(y)\) on linear scales looks the same as \(x\) vs. \(y\) on log scales.

- **training**: During this key stage, the domain of the scale is learned. Sometimes learning the domain of a scale is extremely straightforward: In a plot with only one layer, representing only raw data, it consists of determining the minimum and maximum values of a continuous variable (after transformation), or listing the unique levels of a categorical variable. However, often the domain must reflect multiple layers across multiple datasets in multiple panels. For example, imagine a scale that will be used to create an