标度、坐标轴和图例

简介（1/2）

- **标度（scale）**控制着数据到图形属性的映射。标度将数据转化为视觉上可以感知的东西：大小、颜色、位置、形状。
- 标度也提供了读图时所使用的工具：坐标轴和图例，它们称为**引导元素**（用于允许读者从图形属性空间到数据空间进行反向映射）。
- 每一种**标度**都**是从数据空间**的某个区域（**标度的定义域**）**到图形属性**的某个区域（**标度的值域**）的一个**函数**。

简介（2/2）

- 执行标度的过程分三步：变换（transform）、训练（training）、映射（mapping）。
- 图形中的每一个图形属性都需要一个标度。
- 通常ggplot2将自动添加一个默认的标度。
- 标度分四类：位置标度、颜色标度、手动离散型标度、同一型标度。

标度的工作原理（1/2）

- 标度的**定义域**，即数据空间的值域。输入变量可能是离散型或连续型，则标度的定义域是集合或实值区间。
- 标度的**值域**可以是离散型或连续型。对于离散性标度，其值域是与输入值对应的图形属性值组成的一个向量。对于连续型标度，它的值域是穿过某种个复杂空间的一条一维路径，例如从一种颜色到另一种颜色进行线性插值。

标度的工作原理（2/2）

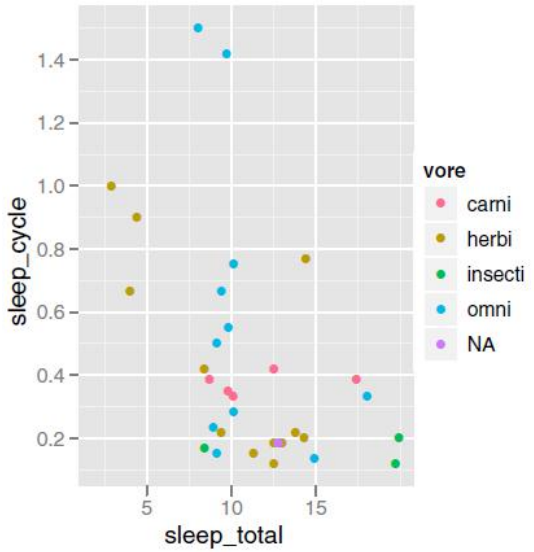
- 将定义域映射到值域的过程包括以下阶段：
  - 变换：针对连续型的定义域，对数据进行变换再展示往往是有益的，比如取对数或开根号。
  - 训练：通过学习（learning）**得到标度的定义域**。定义域必须确保覆盖多个图层上的数据集的变量的范围。
  - 映射：执行映射函数，将数据映射到图形属性。

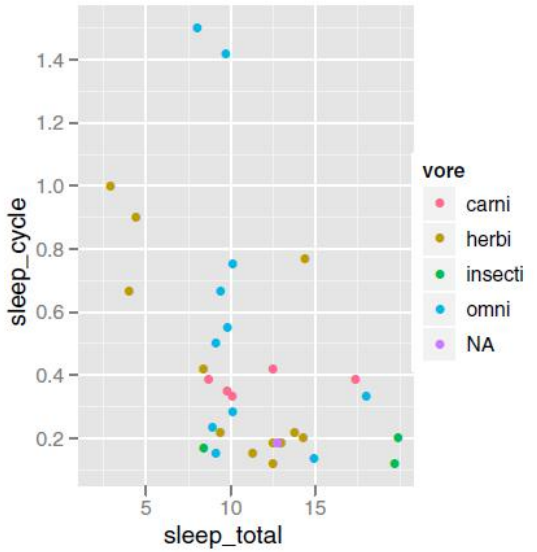
用法

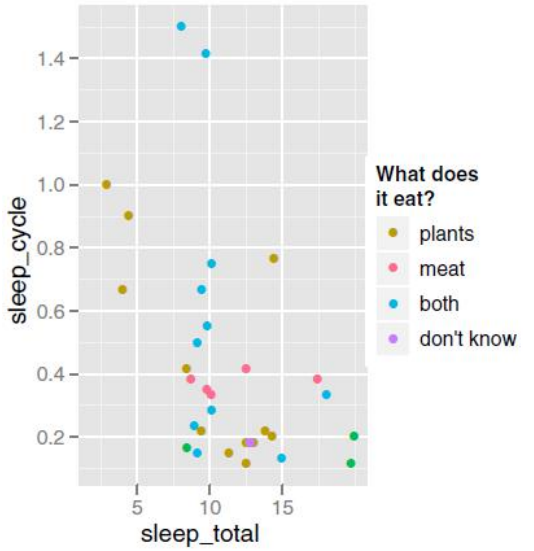
按图形属性和变量类型排列的各种标度

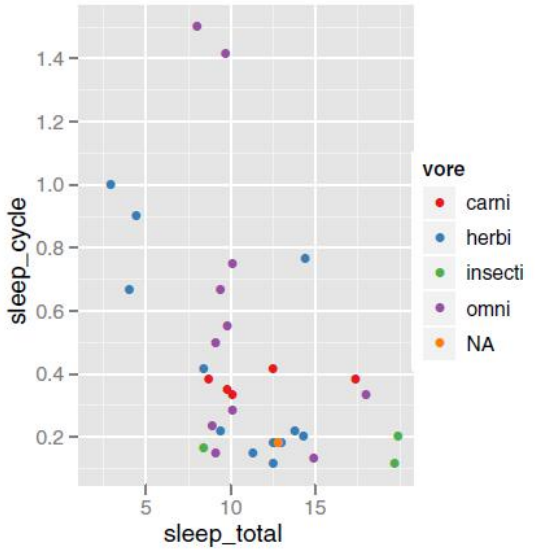
图形属性	离散型	连续型
颜色 (colour) 和填充色 (fill)	brewer	**gradient**
	grey	gradient2
	hue	gradientn
	identity	
	manual	
位置 (position)(x, y)	**discrete**	**continuous**
		date
形状 (shape)	**shape**	
	identity	
	manual	
线条类型 (line type)	**linetype**	
	identity	
	manual	
大小 (size)	identity	**size**
	manual	

例









- p <- qplot(sleep_total, sleep_cycle, data = msleep, colour = vore)
- p #左上图
- p + scale_colour_hue() #右上图，显式添加默认标度，跟左上图一样
- p + scale_colour_hue("What does\nit eat?", breaks = c("herbi", "carni", "omni", NA), labels = c("plants", "meat", "both", "don't know")) #左下图：修改了默认标度的参数：改变了图例的外观。
- p + scale_colour_brewer(palette= "Set1") #右下图：使用了不同的标度：改变了点的颜色

标度详解

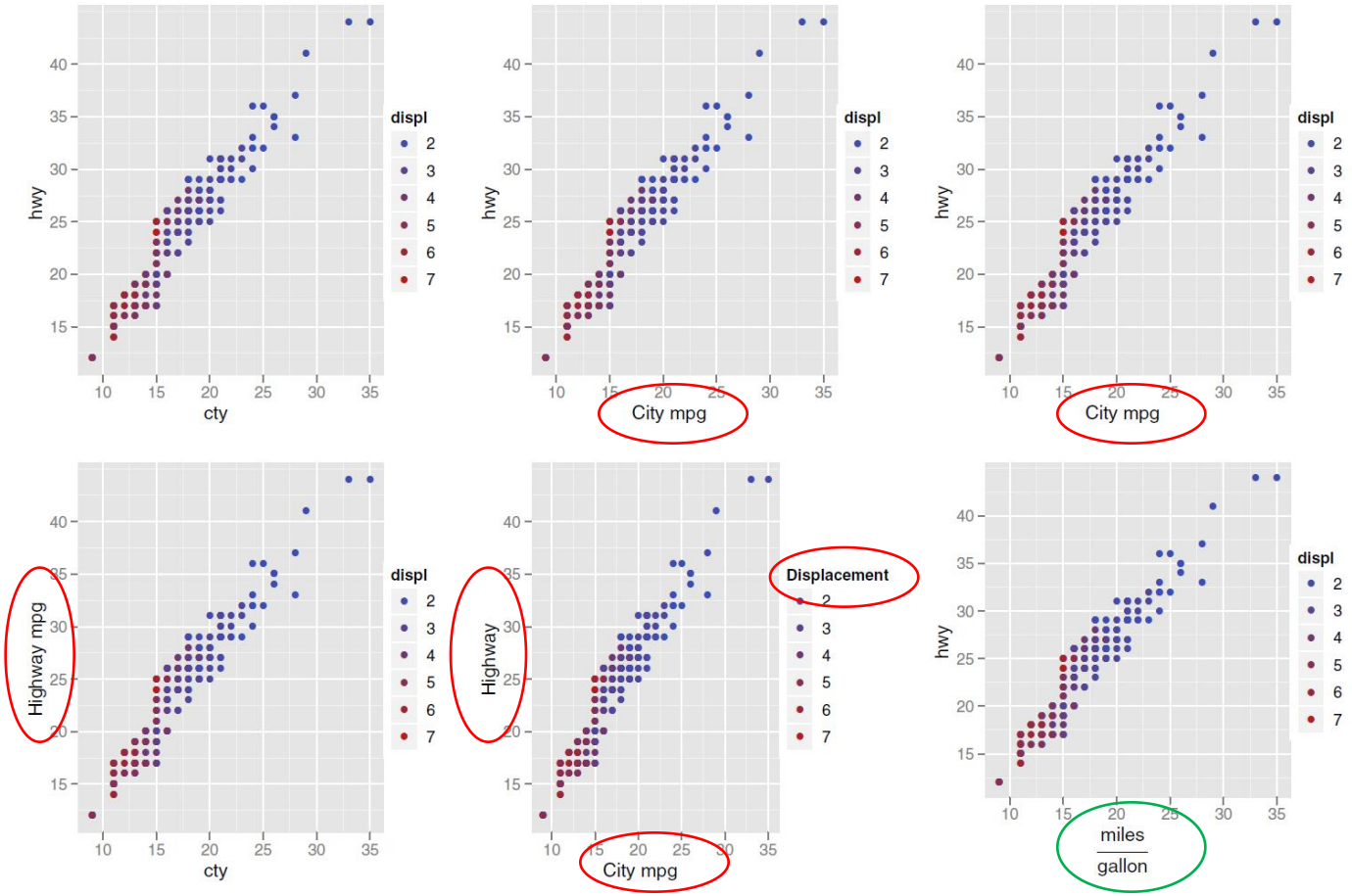
标度详解

- 标度可分为四组：
  - 位置标度
    - 将变量映射到绘图区域，并构造对应的坐标轴。
  - 颜色标度
    - 将变量映射到颜色。
  - 手动离散型标度
    - 奖离散型变量映射到我们选择的符号大小、线条类型、形状、颜色，以及创建对应的图例。
  - 同一型标度
    - 用于直接将变量值绘制为图形，而不去映射它们。这种情况时，变量值本身为图形属性的值。

通用参数（1/2）

- name：坐标轴或图例上出现的标签。可以指定字符串（使用\n换行）或数学表达式。可以使用三个辅助函数xlab()，ylab()，labs()简化编程。
- 例子见下图。

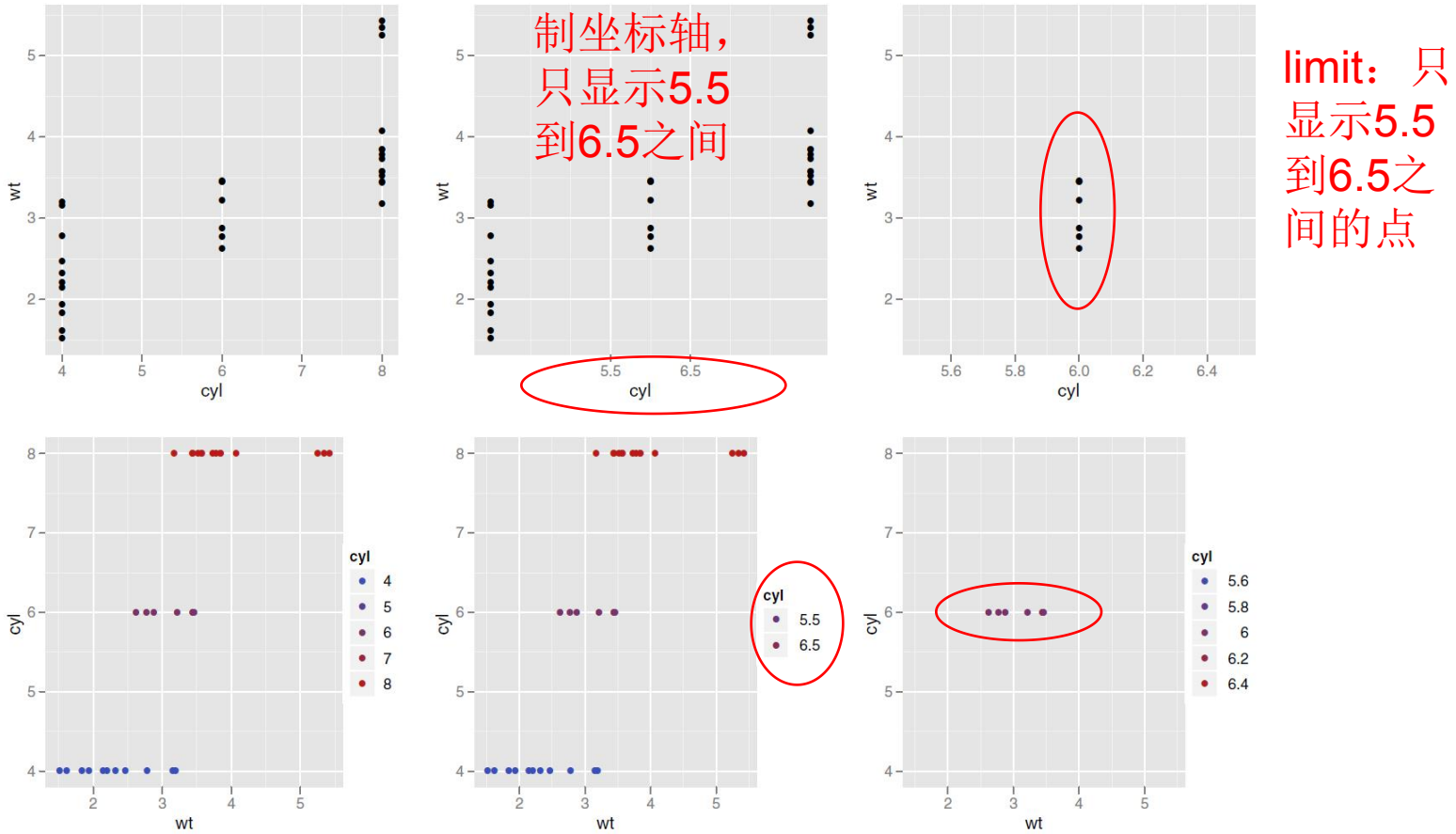
```
p <- qplot(cty, hwy, data = mpg, colour = displ)
p #左上
p + scale_x_continuous("City mpg") #中上
p + xlab("City mpg") #右上
p + ylab("Highway mpg") 左下
p + labs(x = "City mpg", y = "Highway", colour = "Displacement") #中下
p + xlab(expression(frac(miles, gallon))) #右下
```



通用参数（2/2）

- limits：固定标度的定义域。
- breaks和labels：breaks控制着显示在坐标轴或图例上的值。labels指定了应在断点处显示的标签。若设置了labels，则必须同时指定breaks。
- formatter：如果未指定任何标签，则将在每个断点处自动调用格式刷（formatter）来格式化生成标签。
- 部分例子见下页。

```
p <- qplot(cyl, wt, data = mtcars)
p
p + scale_x_continuous(breaks = c(5.5, 6.5)) #中上：breaks控制坐标轴或图例上的值
p + scale_x_continuous(limits = c(5.5, 6.5)) #右上：limit控制显示在图形上的元素
p <- qplot(wt, cyl, data = mtcars, colour = cyl)
p
p + scale_colour_gradient(breaks = c(5.5, 6.5)) #中下：breaks控制坐标轴或图例上的值
p + scale_colour_gradient(limits = c(5.5, 6.5)) #右下：limit控制显示在图形上的元素
```



位置标度

- 每幅图一定有两个位置标度：x标度（水平位置）和y标度（竖直位置）。
- ggplot2提供了连续型、离散型、日期型标度。
- **修改坐标轴范围**是常见任务。可以用xlim()和ylim()来实现。
  - xlim(10, 20): 从10到20的连续型标度
  - ylim(20, 10): 从20到10的反转后连续型标度
  - xlim("a", "b", "c"): 离散型标度
  - xlim(as.Date(c("2008-05-01", "2008-08-01"))): 从2008年5月1日到8月1日的日期型标度

位置标度：连续型

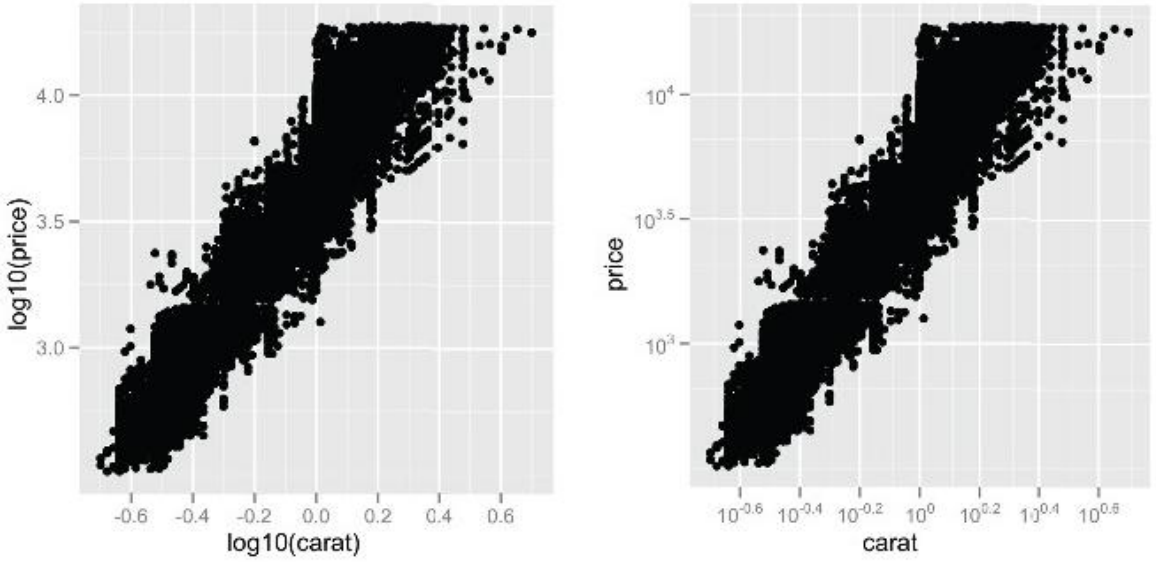
- 最常用的连续型位置标度是scale_x_continuous和scale_y_continuous，它们将数据映射到x轴和y轴。
- 每个连续型标度均可接受一个trans参数，用来指定若干种线性或非线性变换。每一种变换都是由“变换器”（transformer）来实现的。

内建的常用变换器

Name	Function $f(x)$	Inverse $f^{-1}(y)$
asn	$\tanh^{-1}(x)$	$\tanh(y)$
exp	e^x	$\log(y)$
identity	x	y
log	$\log(x)$	e^y
log10	$\log_{10}(x)$	10^y
log2	$\log_2(x)$	2^y
logit	$\log(\frac{x}{1-x})$	$\frac{1}{1+e(y)}$
pow10	10^x	$\log_{10}(y)$
probit	$\Phi(x)$	$\Phi^{-1}(y)$
recip	x^{-1}	y^{-1}
reverse	$-x$	$-y$
sqrt	$x^{1/2}$	y^2

例

- qplot(log10(carat), log10(price), data = diamonds)
- qplot(carat, price, data = diamonds) + **scale_x_log10()** + **scale_y_log10()** **#两图的图形主体是完全相同的，但坐标轴上的标签是不同的。**

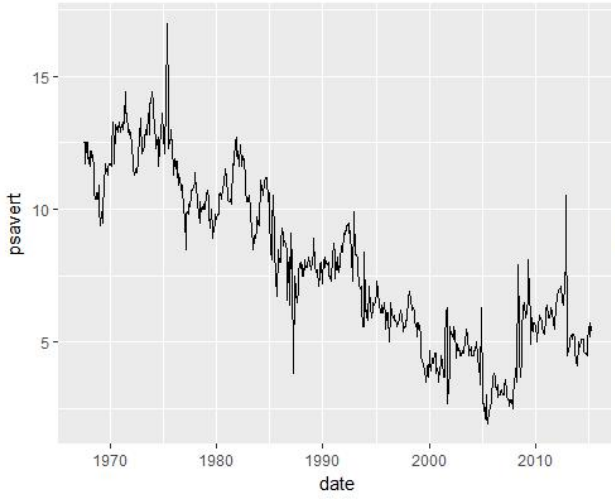


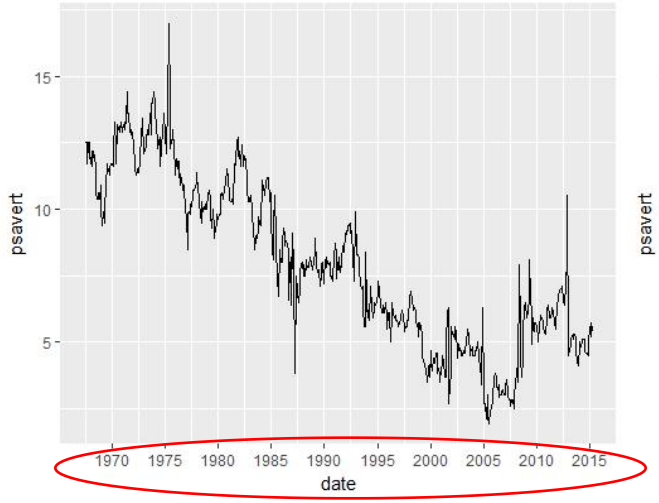
日期和时间（1/3）

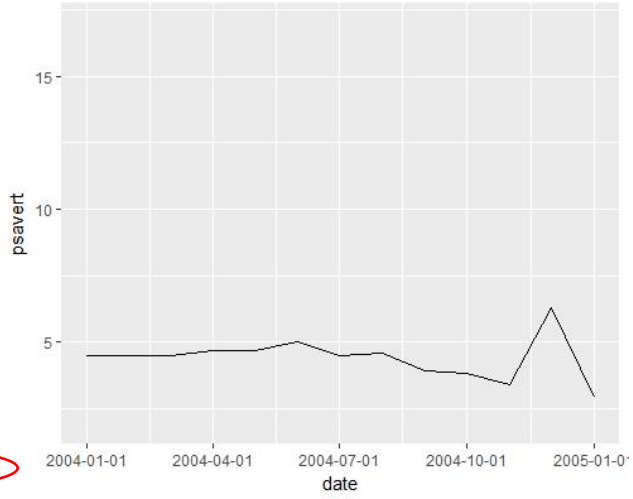
- 日期和时间基本属于连续型。
- 目前标注坐标轴仅支持date类的日期值和POSIXct类的时间值。其他格式需用as.Date()或as.POSIXct()对其进行转换。
- 日期坐标轴，有两个个参数用于控制其外观和刻度的位置。
  - date_breaks()：指定断点位置，并允许以这些单位的倍数出现。如果未指定，日期标度可以自动选出合适的默认值。
  - date_format()：指定了刻度标签的格式。

日期和时间（2/3）

```
library(scales)
plot <- qplot(date, psavert, data = economics, geom = "line") #使用默认设置
plot #左图
plot + scale_x_date(breaks=date_breaks("5 years"), labels =
date_format("%Y")) #中图：设置断点为5年，并把日期格式设置为只显示年
plot + scale_x_date(
    limits = as.Date(c("2004-01-01", "2005-01-01")),
    labels = date_format("%Y-%m-%d") #右图：设置显示区间和日期格式
)
```







日期和时间（3/3）

表：日期格式（在“数据处理”一章中已涉及）

Code	Meaning
%S	second (00-59)
%M	minute (00-59)
%l	hour, in 12-hour clock (1-12)
%I	hour, in 12-hour clock (01-12)
%H	hour, in 24-hour clock (00-23)
%a	day of the week, abbreviated (Mon-Sun)
%A	day of the week, full (Monday-Sunday)
%e	day of the month (1-31)
%d	day of the month (01-31)
%m	month, numeric (01-12)
%b	month, abbreviated (Jan-Dec)
%B	month, full (January-December)
%y	year, without century (00-99)
%Y	year, with century (0000-9999)

离散型

- 离散型位置标度将输入中的各水平映射为整数。
- 结果的顺序可以用参数breaks进行控制，不想要的水平可以用limits（或使用xlim()或ylim()）进行丢弃。

颜色标度

颜色标度（1/3）

- 除了位置标度以外，最常用的就是颜色标度。
- 在物理层面，颜色是由不同波长的光混合的。但由于人类眼球只有三种颜色感受器，可以仅用三个数字来表示任意颜色。
- **rgb**编码的色彩空间的**问题**是在视觉感知上并不均匀，两种间隔一个单位的颜色可能看上去非常相似，但又可能非常不同。

颜色标度（2/3）

- hcl色彩空间（现代方案）有三部分构成：
  - 色相（hue）：0和360之间的值，“颜色”属性：如蓝、红、橙等。
  - 明度（luminance）：颜色的明暗程度。0为黑，1为白。
  - 彩度（chroma）：色彩的纯度。彩度为0是灰色。彩度的最大值随明度的变化而不同。

颜色标度（2/3）

- 约10%的人不具有健全的颜色感受器。因此要避免使用红、绿对比。
- 可以使用模拟色盲的系统来检查图形：
  - Visicheck：在线
  - 使用dichromat包
    - 色盲模拟
    - 色盲配色方案
    - 黑白打印机配色方案

连续型（1/2）

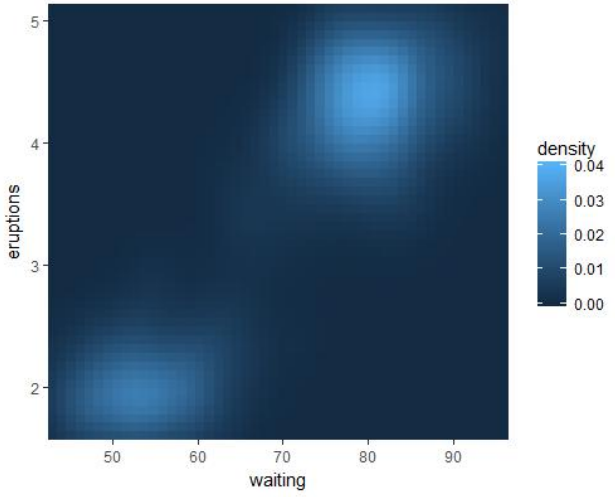
- 根据颜色梯度中的色彩数量划分，共有三类连续型颜色梯度（即渐变色）：
  - scale_colour_gradient()和scale_fill_gradient()
    - 双色梯度。参数low和high控制此梯度两端的颜色。
  - scale_colour_gradient2()和 scale_fill_gradient2()
    - 三色梯度。除了low和high，可以使用参数midpoint设置为任意值（默认是0）。这个参数对生成发散型配色方案特别有用。
  - scale_colour_gradientn()和 scale_fill_gradientn()
    - 自定义的n色梯度。

连续型（2/2）

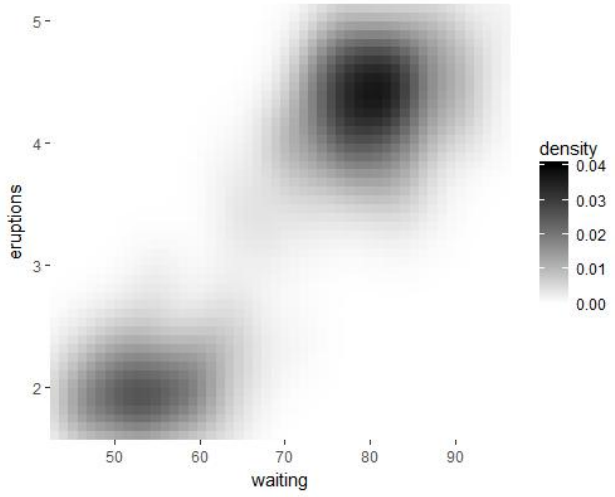
- 颜色梯度常常被用来展示一个二维表面的高度。
- 例：使用faithful数据集（黄石公园的老忠实泉的两次喷发间隔时间以及每次喷发的持续时长）

- f2d <- with(faithful, MASS::kde2d(eruptions, waiting, h = c(1, 10), n = 50))
- df <- with(f2d, cbind(expand.grid(x, y), as.vector(z)))
- names(df) <- c("eruptions", "waiting", "density")
- erupt <- ggplot(df, aes(waiting, eruptions, fill = density)) + geom_tile() + scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))
- erupt + scale_fill_gradient(limits = c(0, 0.04))
- erupt + scale_fill_gradient(limits = c(0, 0.04), low = "white", high = "black")
- erupt + scale_fill_gradient2(limits = c(-0.04, 0.04), midpoint = mean(df$density))

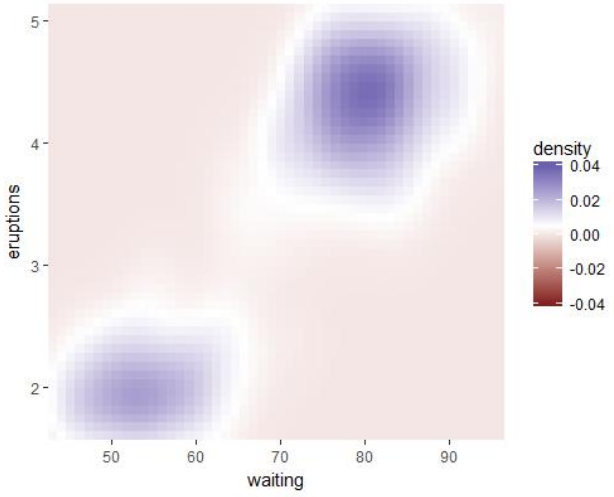
默认的双色梯度



黑白两色的双色梯度



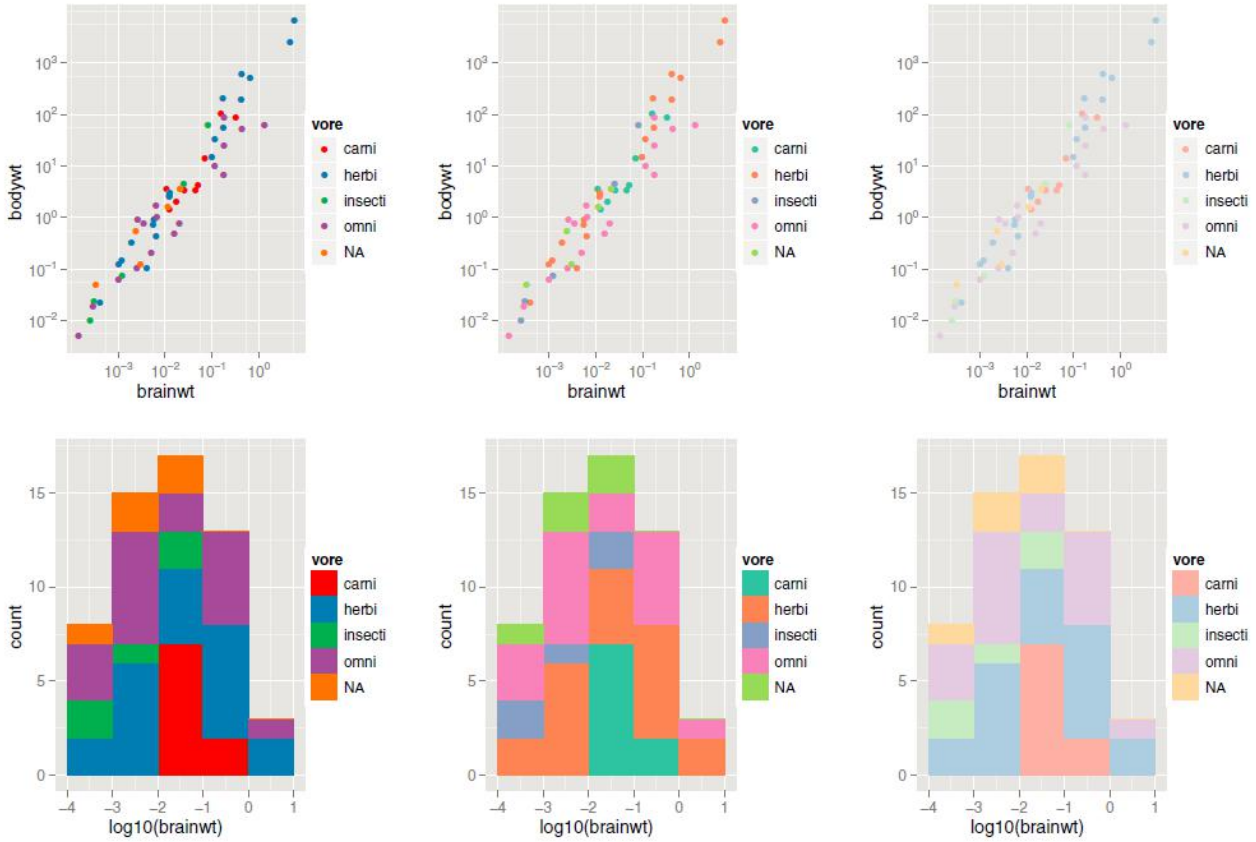
中点设为密度均值的三色梯度



离散型

- 离散型默认的配色方案，即scale_colour_hue()，通过沿着hcl色轮选取均匀分布的色相来生成颜色。这种方案对至多约8种颜色时都能有较好效果。默认配色进行黑白打印时，由于所有颜色拥有相同的明度和彩度，会成为几乎相同的灰影。
- 另一种可选的方案是使用ColorBrewer配色。
  - 例子见下页。

```
point <- qplot(brainwt, bodywt, data = msleep, log = "xy", colour = vore)
area <- qplot(log10(brainwt), data = msleep, fill = vore, binwidth = 1)
point + scale_colour_brewer(palette = "Set1") #左上图
point + scale_colour_brewer(palette = "Set2") #中上图
point + scale_colour_brewer(palette = "Pastel1") #右上图
area + scale_fill_brewer(palette = "Set1") #左下图
area + scale_fill_brewer(palette = "Set2") #中下图
area + scale_fill_brewer(palette = "Pastel1") #右下图
```



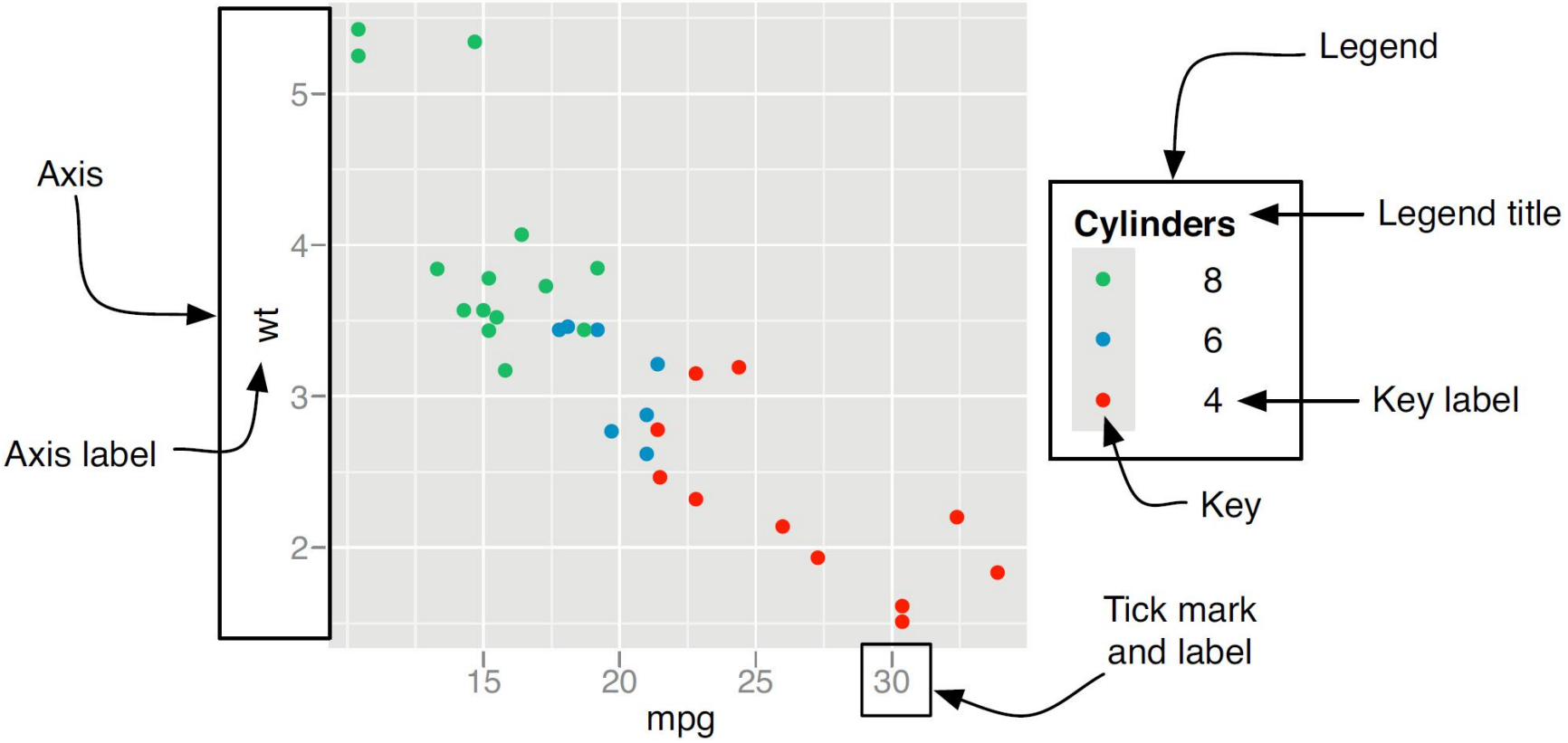
三种ColorBrewer调色板：Set1（左两图）、Set2（中两图）、Pastel1（右两图）。更明亮的颜色对点效果良好，但对于条形图来说过于刺眼。淡色对条形图效果良好，但会让点看不清楚。

图例和坐标轴

图例和坐标轴

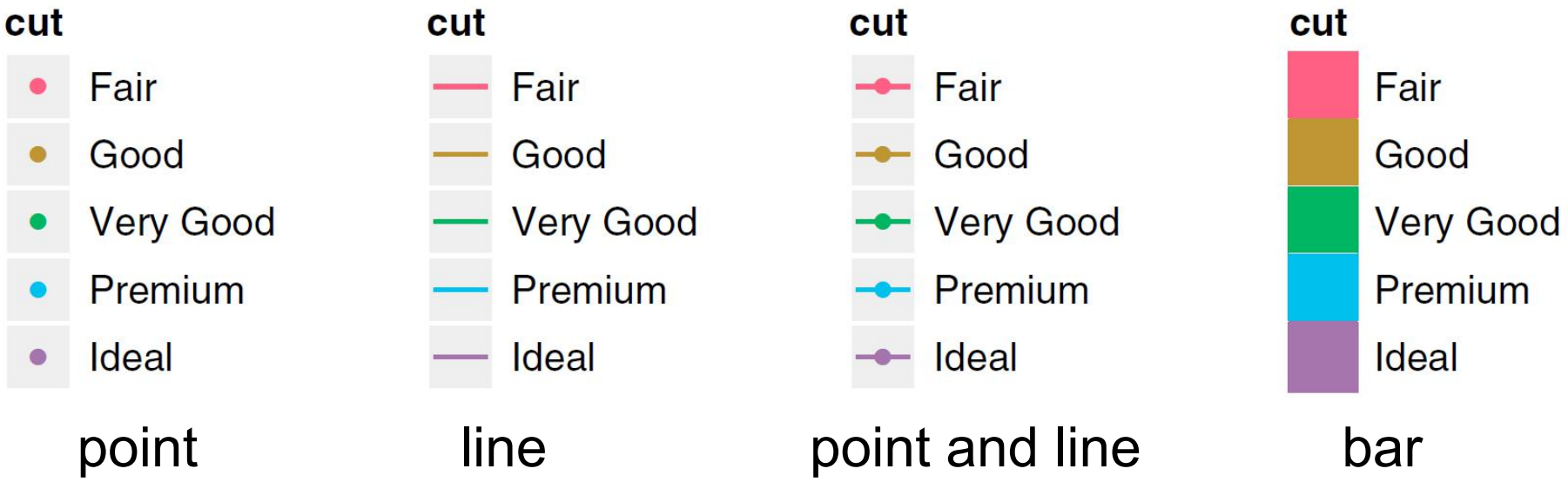
- 图例和坐标轴被共同成为**引导元素**，它们都是标度的逆函数：它们允许你在图中读出观测并将其映射为原始值。
- 图例和坐标轴很相似：图例标题（legend title）和坐标轴名（axis lebel）是等价的，皆由标度的名称参数（name）决定；图例标示和刻度标签皆由断点参数（break）决定。

坐标轴和图例的组成部分



绘制图例（1/2）

- 要绘制图例，图形必须收集**每一种图形属性**的使用信息：为何种数据以及为何种几何对象。标度的断点（**breaks**）被用来**确定**图例**标示的值**。对应图形属性的**几何对象确定**如何绘制**标示**。



图：由不同几何对象生成的图例

绘制图例（2/2）

- ggplot2会生成最小数量的能够表达图中使用图形属性的图例。当一个变量对应多个图形属性时，可以通过合并图例的方法来精简。例：如果颜色（左图）和形状（中图）都被映射到相同的变量，那么用一个图例（右图）就够了。