```
andrews_curve(iris[, 1:4], n = 50, col = iris.col,
              xlab = "t", ylab = "f(t)")
legend("topleft", col = unique(iris.col), lty = 1, bty = "n",
       legend = unique(iris$Species))
andrews_curve(iris[, c(3, 4, 2, 1)], n = 50, col = iris.col,
              xlab = "t", ylab = "f(t)")
andrews_curve(scale(iris[, 1:4]), n = 50, col = iris.col,
              xlab = "t", ylab = "f(t)")
x = andrews_curve(scale(trees), n = 50,
                  xlab = "t", ylab = "f(t)")
```

离群点是哪行数据？即哪行数据对应的 $f(t)$ 会大于 4？很简单：

```
which(apply(x > 4, 1, any))
```

```
## [1] 31
```

注意：`andrews_curve()` 函数会返回所有行（每条观测）在每个 t 值上对应的 $f_i(t)$ 值，我们可以根据这个返回值来判断图中各条曲线对应的行。

6.13 地图

概述

毫无疑问，地图是展示地理信息数据时最直观的工具，尤其是当地图和统计量结合时，其功效则会进一步加强。在本书的第一章中曾经提到过 John Snow 的地图，注意图中不仅标示出了霍乱发生的地点，每个地点的死亡人数也用点的数目标示了出来。历史上还有不少类似的使用地图的例子，而在今天，地理信息系统（GIS）已经成为研究空间和地理数据的热门工具，地图的应用也是屡见不鲜。

示例

表 6.10 给出了 2005 年世界各国家地区的农业进出口竞争力指标数据 [Xie, 2007]。其中，我们将竞争力指标简单定义为（出口 − 进口）/（出口 + 进口）。我们将这组数据在图 6.15 上标示出来。从图中可以看出，阿根廷、巴西等南美国家的农业进出口竞争力较强，而利比亚、阿尔及利亚等北非国家的竞争力较弱。

表 6.10: 2005 年世界各国农业进出口竞争力（部分。原数据为 97 行 2 列）

Index	Country
-0.7570701	Albania
-0.9667213	Algeria
0.8778870	Argentina
0.5264728	Australia
0.0062974	Austria

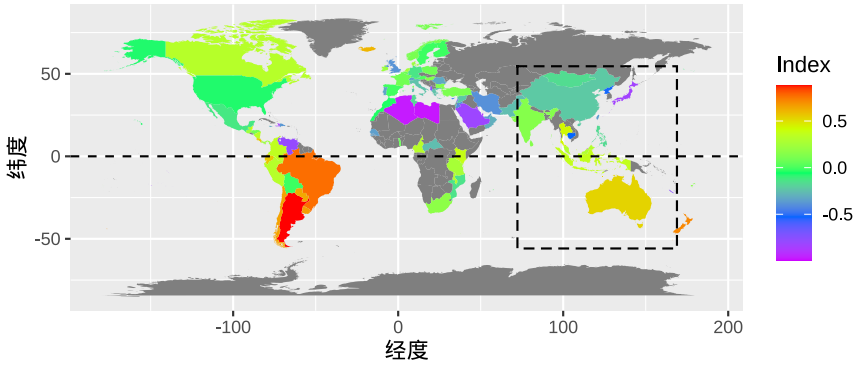


图 6.15: 2005 年各国农业进出口竞争力地图：农业出口强国在南美，弱国在北非

绘制方法

地图的本质是多边形（9.2.5 小节），而多边形的边界则由地理经纬度数据确定。R 中绘制地图的传统附加包是 **maps** [code by Richard A. Becker et al., 2018]，核心的函数为 `map()`，它的用法如下：

```
map(database = "world", regions = ".", exact = FALSE, boundary = TRUE,
    interior = TRUE, projection = "", parameters = NULL,
    orientation = NULL, fill = FALSE, col = 1, plot = TRUE,
    add = FALSE, namesonly = FALSE, xlim = NULL, ylim = NULL,
```

```
    wrap = FALSE, resolution = if (plot) 1 else 0, type = "l",
    bg = par("bg"), mar = c(4.1, 4.1, par("mar")[3], 0.1),
    myborder = 0.01, namefield = "name", lforce = "n", ...)
```

该函数的两个主要参数为地图数据库 database 和地图区域 region。地图数据库中包含了所有区域的经纬度数据以及相应的区域名称。在指定一个数据库和一系列区域名称之后，这些区域的地图便可由 maps() 生成。其它参数诸如填充颜色、是否画边界、是否添加到现有图形上等这里就不再介绍，请读者参考帮助文件。

运行下面的代码可以将表 6.10 的数据绘制成地图：

```
# 世界各国农业进出口竞争力地图
source(system.file("extdata", "AgriComp.R", package = "MSG"))
demo("AgriComp", package = "MSG")
```

上述代码的大致制作过程为：首先我们用 world 数据库作出一幅空白的世界地图，地区边界用灰色线条表示，然后我们根据竞争力数据中的地区名称与地理数据库中地区名称的对应将数据以颜色的形式表示到世界地图中，最后我们在图中添加了赤道线以及东盟国家（ASEAN）的矩形区域，这是由于作为该图出处的会议论文 [Xie, 2007] 主题是中澳自由贸易区。

maps 包功能虽然比较完善，但绘制地图的过程仍然有些繁琐。近年来，**ggplot2** 发展迅猛，其地图绘制功能也越来越强大了。图 6.15 实际是用 **ggplot2** 包绘制多边形的方式作出来的，代码如下：

```
# ggplot2 绘制世界各国农业进出口竞争力地图
source(system.file("extdata", "AgriComp.R", package = "MSG"))
library(ggplot2)
worldmap = map_data("world")
worldmap2 = dplyr::left_join(worldmap, AgriComp,
                             by = c("region" = "Country"))
p = ggplot(worldmap2) +
  geom_polygon(aes(long,lat, group=group,  fill = Index)) +
  coord_quickmap() +
  scale_fill_gradientn(colours = rev(rainbow(5))) +
  labs(x = "经度", y = "纬度") +
  geom_rect(data = data.frame(xmin = 72.26818, ymin = -55.8565,
                              xmax = 168.93766, ymax =54.589),
```

```
            mapping = aes(xmin = xmin, ymin = ymin,
                          xmax = xmax, ymax =ymax),
            fill = NA, color = "black", linetype = 2) +
  geom_hline(yintercept = 0, linetype = 2)
print(p)
```

这里，coord_quickmap() 函数，专门用作地图坐标，确保地图上的经度和纬度之比符合常用的摩克托投影规则。此外，**ggplot2** 还提供了另外一个坐标转换函数 coord_map()，功能更复杂一些。如图 6.16 所示，上图是平面地图，下图是以北纬 20° 东经 90° 视角看到的球状地图。生成这两幅地图的代码如下：

```
# 世界地图的两个视角
library(ggplot2)
library(patchwork)
m0 = ggplot() +
  geom_polygon(data = worldmap,
               mapping = aes(long,lat, group=group,  fill = region)) +
  guides(fill = FALSE)
m1 = m0 + coord_quickmap()
m2 = m0 + coord_map("ortho", orientation = c(20,90,0))
print(m1 / m2)
```

在地理区域上标记大量的数值信息会遇到一个显而易见的困难，就是由于各个地理区域的面积不同而导致地图的解读失真或某些重要地理单元难以辨认。例如，我们在画中国省级地图时，北京和上海等直辖市相比其它省份显得面积太小，此时若用颜色来标记某个数值指标（如 GDP），就会使得各个直辖市的颜色几乎无法辨认。

一个更有趣的例子来自 2008 年美国总统大选，如图 6.17。若用红蓝两种颜色对各个州做标记，以表示该州支持麦凯恩或奥巴马，那么有些面积不大但是权重很大的州（如人口众多的加州）就会影响整幅美国地图。从原始地图上看，似乎麦凯恩会赢，因为他赢得了很多中部面积大的州（但人口稀少），整幅地图看起来以红色为主导；若我们保持州的相对地理位置不变，将各个州的形状进行大小的调整，使其面积与权重成正比，此时红蓝两色的局面就发生了逆转，地图以蓝色为主导色，地图传达信息的偏误才得到了纠正。我们把这种保持地理区域的相对位置不变、调整区域面积与某指标成比例的地图成为“变形地图”（Cartogram），详

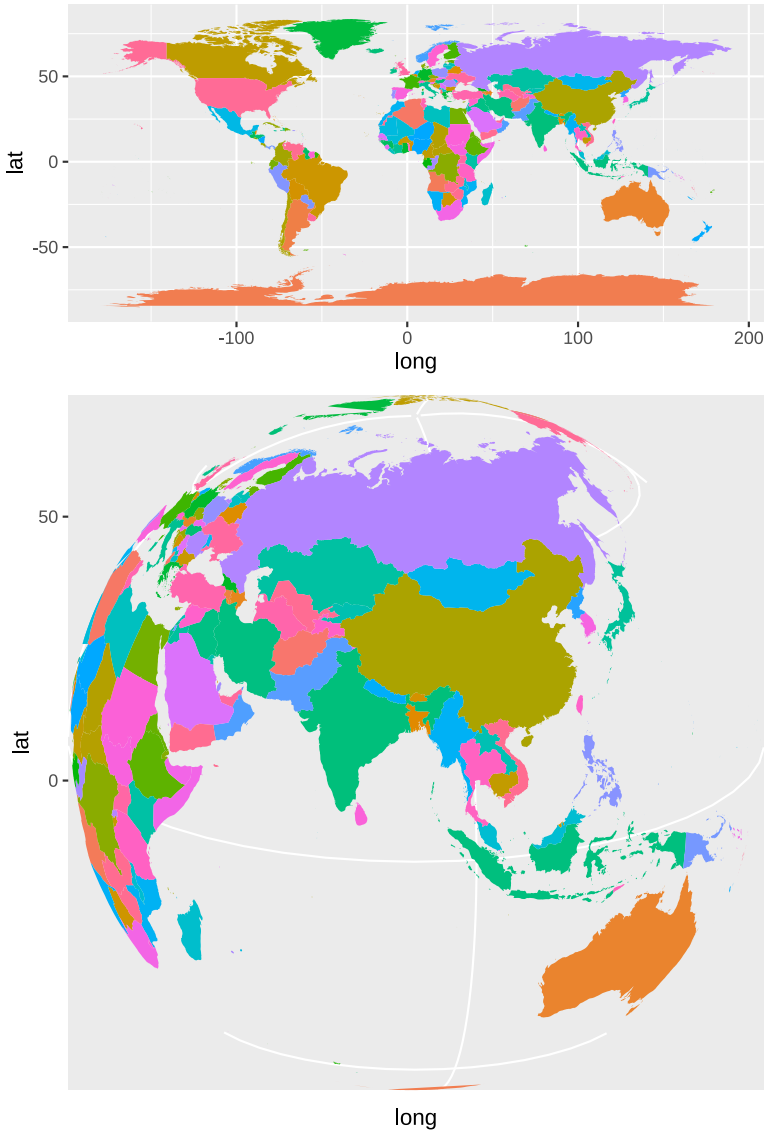


图 6.16: 不同视角和投影下的世界地图：上图是平面地图，下图是以北纬 20° 东经 90° 视角看到的球状地图

细内容可阅读笔者的博客[2]。

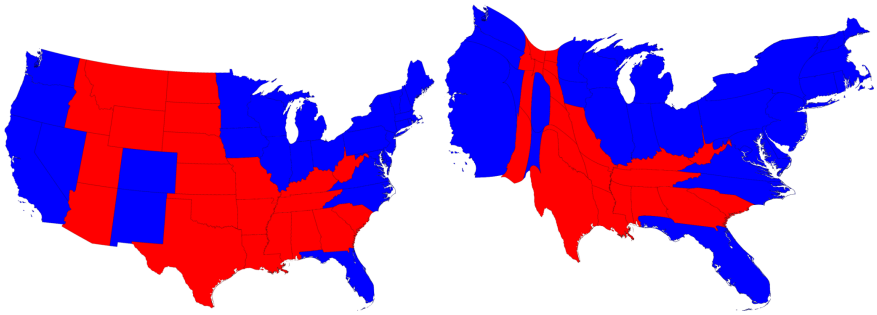


图 *6.17: 2008* 年美国总统大选各州投票情况：红蓝两种颜色分别表示该州支持麦凯恩或奥巴马。左图是正常地图，麦凯恩赢得了很多中部面积大的州（但人口稀少），整幅地图看起来以红色为主导；右图为变形地图，保持了地理区域的相对位置不变而调整区域面积与权重成比例，地图以蓝色为主导色

除了 **maps** 包和 **ggplot2** 包之外，R 语言还有更多的地图绘制系统，例如 **RgoogleMaps** 包 [Loecher and Ropkins, 2015]，将 Google Maps 提供的（卫星）地图数据引入 R 中，这里简单介绍一下。

首先，此包利用 Google Maps API，为 R 提供了一个十分便利的接口，以抓取 Google 服务器上的静态地图；其次，用户可使用获得的地图作为背景，在其上方自由叠加图形元素。对于一般的经纬度坐标数据，此包可计算包含这些数据点的矩形边界，以确定抓取地图的范围。其工作流程概括如下：

- 读取经纬度数据
- 通过计算确定获取图片所需参数
- 访问 Google Maps 服务器抓取图片
- 依据经纬度数据在图片上叠加图形元素

下面我们举一个利用 **RgoogleMaps** 包的例子。

表 6.11 给出了来自中国国家地震科学数据共享中心的 354 条四川地区地震数据示例。3 个变量分别为震源的纬度、经度和震级大小（单位：面波震级 Ms），时间跨度为 2010 年 3 月 23 日到 2010 年 4 月 23 日。

[2] https://yihui.org/cn/2009/03/cartogram-as-special-maps/

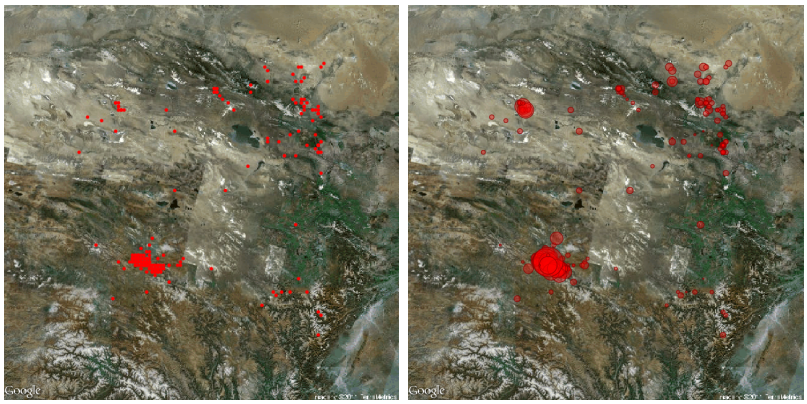
表 6.11: 四川地区地震数据（部分。原数据为 354 行 3 列）

lat	long	ms
33.2	96.6	1.180
37.5	102.8	1.067
32.3	101.5	0.276
33.1	96.7	1.067
33.3	96.3	1.180

图 6.18 显示了地震震源位置分布情况，背景采用了 Google Maps 提供的卫星地图数据。左图仅仅体现了震源位置的分布情况，不妨考虑将震级的大小映射为圆的半径大小。然而，图中存在着部分地震多发地带，如果使用圆来呈现震源的位置，这些区域的圆将出现严重的叠加现象，此处可以尝试使用 9.1.1 节中的透明度叠加来克服这类重叠问题，如右图所示。这里由于数据量不够大，这种透明度叠加的效果并不是非常明显。

本节具体的代码参见 eqMaps 演示：

```
# 在卫星地图上标记地震发生的地点和震级
demo("eqMaps", package = "MSG")
```



(a) (b)

图 6.18: 在卫星地图上标记地震发生的地点和震级：左图仅标记地点，右图用圆圈大小代表震级大小

RgoogleMaps 包的潜力仍尚待挖掘。一方面，它可以展示空间分

布信息，例如在 2010 年的 **ggplot2** 案例分析竞赛中，David Kahle 利用 **RgoogleMaps** 包和公开的犯罪信息数据，展示了休斯顿地区暴力犯罪的分布情况[3]；另一方面，如果数据包含时间属性，那么我们可以固定住抓取图片的边界，并保证叠加元素的坐标对应正确，便能制作出有用的动画。读者可以发挥想象力，拓展更多的应用情境。

本节只是介绍了一个非常简单的应用，但也引出了一个重要话题：统计图形如何与它要表达的问题的背景相融合？用通俗的话讲，就是要找“应景”的背景。在这方面，图 3.1 实际上做得很好，很有吸引眼球的效果，让人一看就明白要表达的主题。当然，背景元素也不能喧宾夺主，这一点在 8.2.1 小节中有详细论述。

6.14 思考与练习

1. 自行编写一个画三元图的函数，并体会这种从三维到二维的变换。以下是不完整的代码，核心部分已经完成，需要实现控制边长范围和坐标网格线等功能：

```
triplot = function(x, ...) {
  x = as.matrix(x)
  x = x / rowSums(x) # 将行之和标准化到 1
  plot(x[, 2] + x[, 3] / 2, x[, 3] * sqrt(3) / 2, asp = 1,
       ann = FALSE, axes = FALSE, xlim = c(0, 1),
       ylim = c(0, sqrt(3) / 2), ...)
  polygon(c(0, 1, 1 / 2), c(0, 0, sqrt(3) / 2))
}
# 测试数据
data(murcia, package = "MSG")
triplot(murcia[, 2:4], col = vec2col(murcia$site), pch = 19)
```

2. **ggplot2** 并没有现成的函数来绘制条件分割图。不过，下面的代码可以做出类似的效果，如图 6.19。比较一下，这幅图与图 6.3 的区别在哪里？能得出相同的结论吗？如何更改这段代码，以做出图 6.3？

```
# ggplot2 绘制地震经纬度条件分割图
data(quakes)
library(ggplot2)
```

[3] http://1t.click/Ksb