

对于散点图，我已经介绍非常多了，之所以论述这么多的细节，主要是因为它在数据分析中占据着非常重要的位置。虽然散点图很简单，但是它们能帮你以最直接的方式展示数据，发现隐藏的可能会被忽略的关系。

11.2 折线图

如果将散点图上的点从左往右连接起来，那么就会得到一个折线图。以基础安装中的Orange数据集为例，它包含五种橘树的树龄和年轮数据。现要考察第一种橘树的生长情况，绘制图形11-17。左图为散点图，右图为折线图。可以看到，折线图是一个刻画变动的优秀工具。

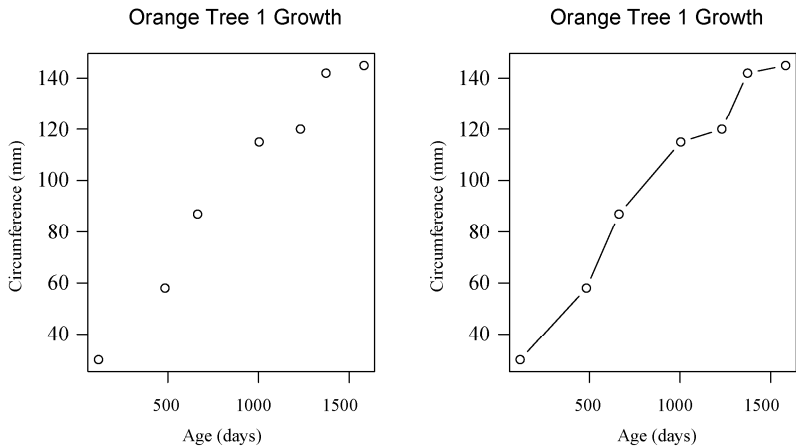


图11-17 散点图与折线图的对比

图11-17是由代码清单11-3中的代码创建的。

代码清单11-3 创建散点图和折线图

```
opar <- par(no.readonly=TRUE)
par(mfrow=c(1,2))
t1 <- subset(Orange, Tree==1)
plot(t1$age, t1$circumference,
     xlab="Age (days)",
     ylab="Circumference (mm)",
     main="Orange Tree 1 Growth")
plot(t1$age, t1$circumference,
     xlab="Age (days)",
     ylab="Circumference (mm)",
     main="Orange Tree 1 Growth",
     type="b")
par(opar)
```

在第3章中，代码中的基本参数你都已经见过，因此此处不做过多讲解。图11-17中两幅图的主要区别取决于参数 `type = "b"`。折线图一般可用下列两个函数之一来创建：

```
plot(x, y, type=)
lines(x, y, type=)
```

其中，x和y是要连接的(x, y)点的数值型向量。参数type =的可选值见表11-1。

表11-1 折线图类型

类 型	图形外观
p	只有点
l	只有线
o	实心点和线（即线覆盖在点上）
b、c	线连接点（c时不绘制点）
s、S	阶梯线
h	直方图式的垂直线
n	不生成任何点和线（通常用来为后面的命令创建坐标轴）

图11-18给出了各类型的示例。可以看到，type = "p"生成了典型的散点图，type = "b"是最常见的折线图。b和c间的不同之处即点是否出现或者线之间是否有空隙。type = "s"和type = "S"都生成阶梯线（阶梯函数），但一种类型是先横着画线，然后再上升，而第二种类型是先上升，再横着画线。

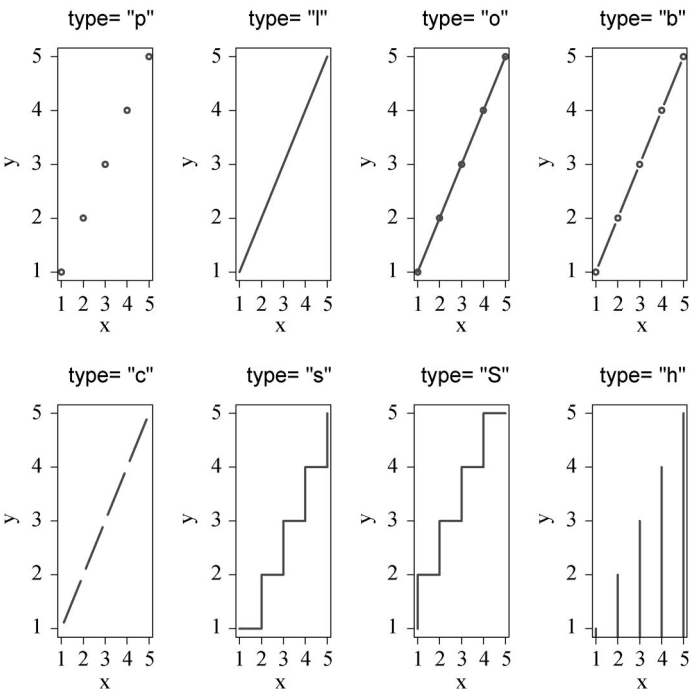


图11-18 plot()和lines()函数中的type参数值

注意, `plot()` 和 `lines()` 函数工作原理并不相同。`plot()` 函数是被调用时即创建一幅新图, 而 `lines()` 函数则是在已存在的图形上添加信息, 并不能自己生成图形。

因此, `lines()` 函数通常是在 `plot()` 函数生成一幅图形后再被调用。如果对图形有要求, 你可以先通过 `plot()` 函数中的 `type = n` 来创建坐标轴、标题和其他图形特征, 然后再使用 `lines()` 函数添加各种需要绘制的曲线。

我们以绘制五种橘树随时间推移的生长状况为例, 逐步展示一个更复杂折线图的创建过程。每种树都有自己独特的线条。代码见代码清单11-4, 结果见图11-19。

代码清单11-4 展示五种橘树随时间推移的生长状况的折线图

```
Orange$Tree <- as.numeric(Orange$Tree)
ntrees <- max(Orange$Tree)

xrange <- range(Orange$age)
yrange <- range(Orange$circumference)

plot(xrange, yrange,
      type="n",
      xlab="Age (days)",
      ylab="Circumference (mm)"
)

colors <- rainbow(ntrees)
linetype <- c(1:ntrees)
plotchar <- seq(18, 18+ntrees, 1)

for (i in 1:ntrees) {
  tree <- subset(Orange, Tree==i)
  lines(tree$age, tree$circumference,
        type="b",
        lwd=2,
        lty=linetype[i],
        col=colors[i],
        pch=plotchar[i]
  )
}

title("Tree Growth", "example of line plot")

legend(xrange[1], yrange[2],
       1:ntrees,
       cex=0.8,
       col=colors,
       pch=plotchar,
       lty=linetype,
       title="Tree"
)
```

← 为方便起见, 将因子转化为数值型

创建图形

添加线条

添加图例

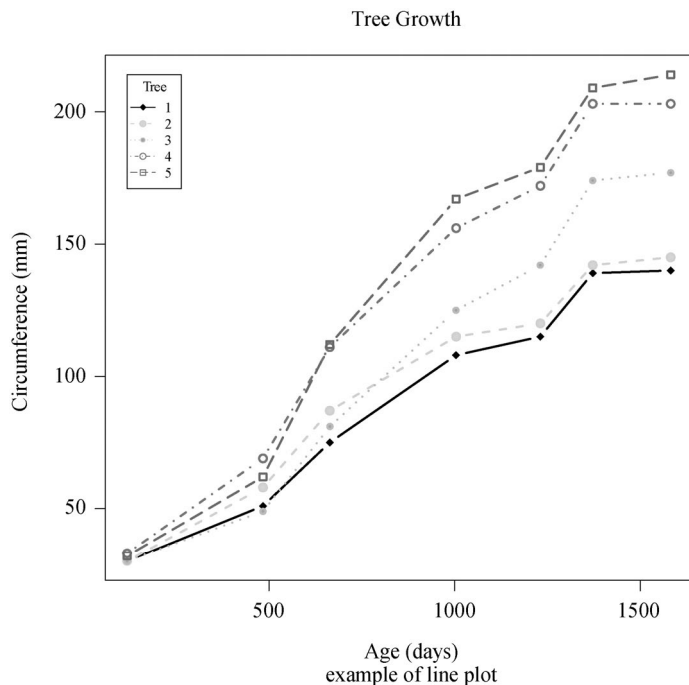


图11-19 展示五种橘树生长状况的折线图（另见彩插图11-19）

在代码清单11-4中，`plot()`函数先用来创建空图形，只设定了轴标签和轴范围，并没有绘制任何数据点，每种橘树独有的折线和点都是随后通过`lines()`函数来添加。可以看到，Tree 4和Tree 5在整个时间段中一直保持着最快的生长速度，而且Tree 5在大约664天的时候超过了Tree 4。

代码清单11-4使用了许多R中的编程惯例，这些惯例在第2章、第3章和第4章都已讨论过。通过亲手一行一行地敲入代码，观察可视化结果，你可以检验是否对这些惯例有了深刻的理解。如果答案是肯定的，那么恭喜你，你正在成为严肃的R程序员（声名和机遇都唾手可得了）！在下一节中，我们将会探索各种同时检验多个相关系数的方法。

11

11.3 相关图

相关系数矩阵是多元统计分析的一个基本方面。哪些被考察的变量与其他变量相关性很强，而哪些并不强？相关变量是否以某种特定的方式聚集在一起？随着变量数的增加，这类问题将变得更难回答。相关图作为一种相对现代的方法，可通过对相关系数矩阵的可视化来回答这些问题。

相关图非常容易解释，你只要看到它就会立马明白。以`mtcars`数据框中的变量相关性为例，它含有11个变量，对每个变量都测量了32辆汽车。利用下面的代码，你可以获得该数据的相关系数：