

本例所得结果因平台而异，因为不同系统中映射的常规字体、等宽字体和有衬线字体有所不同。在你的系统上，结果看起来如何呢？

数学标注

最后，你可以使用类似于TeX中的写法为图形添加数学符号和公式。请参阅`help(plotmath)`以获得更多细节和示例。要即时看效果，可以尝试执行`demo(plotmath)`。部分运行结果如图3-13所示。函数`plotmath()`可以为图形主体或边界上的标题、坐标轴名称或文本标注添加数学符号。

Arithmetic Operators		Radicals	
$x + y$	$x + y$	\sqrt{x}	\sqrt{x}
$x - y$	$x - y$	$\sqrt{x, y}$	$\sqrt[3]{x}$
$x * y$	xy	Relations	
x/y	x/y	$x == y$	$x = y$
$x \%+ \% y$	$x \pm y$	$x != y$	$x \uparrow y$
$x \%/% y$	$x \sqrt{y}$	$x < y$	$x < y$
$x \%* \% y$	$x \times y$	$x <= y$	$x'' y$
$x \%.% y$	$x \cdot y$	$x > y$	$x > y$
$-x$	$-x$	$x >= y$	$x \geq y$
$+x$	$+x$	$x \% \sim \% y$	$x \oplus y$
Sub/Superscripts		$x \% \sim \% y$	$x \equiv y$
$x[i]$	x_i	$x \% == \% y$	$x \equiv y$
x^2	x^2	$x \%prop \% y$	$x \propto y$
Juxtaposition		Typeface	
$x * y$	xy	<code>plain(x)</code>	x
<code>paste(x,y,z)</code>	xyz	<code>italic(x)</code>	x
Lists		<code>bold(x)</code>	x
<code>list(x,y,z)</code>	x, y, z	<code>bolditalic(x)</code>	x
		<code>underline(x)</code>	x

图3-13 `demo(plotmath)` 的部分结果

同时比较多幅图形，我们通常可以更好地洞察数据的性质。所以，作为本章的结尾，下面讨论将多幅图形组合为一幅图形的方法。

3.5 图形的组合

在R中使用函数`par()`或`layout()`可以容易地组合多幅图形为一幅总括图形。此时请不要担心所要组合图形的具体类型，这里我们只关注组合它们的一般方法。后续各章将讨论每类图形的绘制和解读问题。

你可以在`par()`函数中使用图形参数`mfrow=c(nrows, ncols)`来创建按行填充的、行数为`nrows`、列数为`ncols`的图形矩阵。另外,可以使用`nfcol=c(nrows, ncols)`按列填充矩阵。

举例来说,以下代码创建了四幅图形并将其排布在两行两列中:

```
attach(mtcars)
opar <- par(no.readonly=TRUE)
par(mfrow=c(2,2))
plot(wt,mpg, main="Scatterplot of wt vs. mpg")
plot(wt,disp, main="Scatterplot of wt vs disp")
hist(wt, main="Histogram of wt")
boxplot(wt, main="Boxplot of wt")
par(opar)
detach(mtcars)
```

结果如图3-14所示。

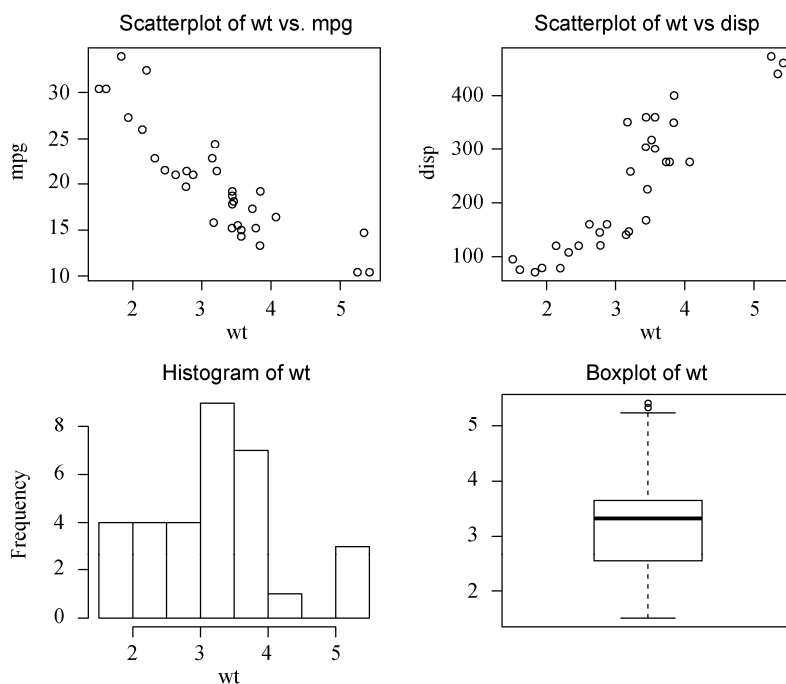


图3-14 通过`par(mfrow=c(2,2))`组合的四幅图形

作为第二个示例,让我们依3行1列排布3幅图形。代码如下:

```
attach(mtcars)
opar <- par(no.readonly=TRUE)
par(mfrow=c(3,1))
hist(wt)
hist(mpg)
hist(disp)
par(opar)
detach(mtcars)
```

所得图形如图3-15所示。请注意，高级绘图函数`hist()`包含了一个默认标题（使用`main=""`可以禁用它，抑或使用`ann=FALSE`来禁用所有标题和标签）。

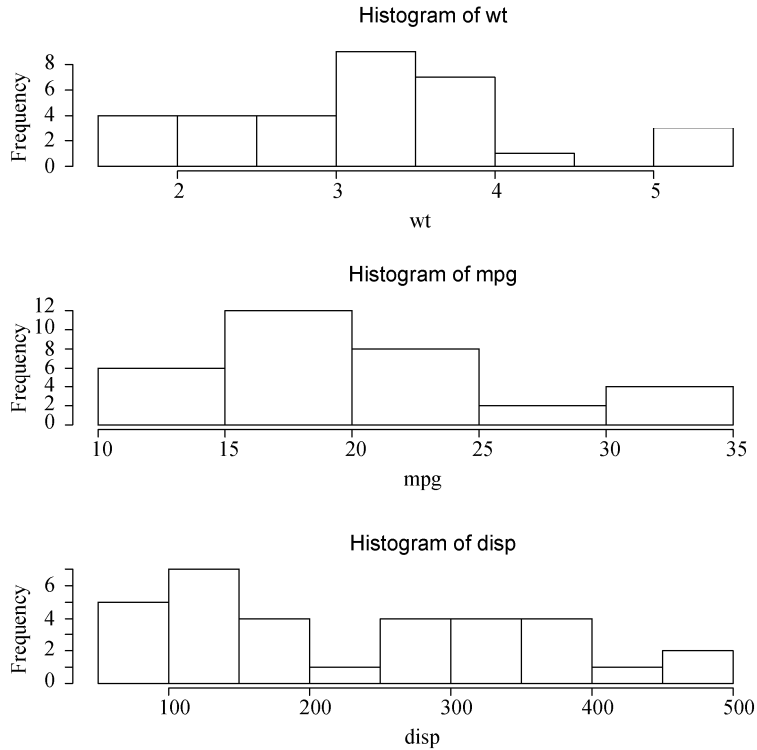


图3-15 通过`par(mfrow=c(3,1))`组合的三幅图形

函数`layout()`的调用形式为`layout(mat)`，其中的`mat`是一个矩阵，它指定了所要组合的多个图形的所在位置。在以下代码中，一幅图被置于第1行，另两幅图则被置于第2行：

```
attach(mtcars)
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))
hist(wt)
hist(mpg)
hist(dis)
detach(mtcars)
```

结果如图3-16所示。

为了更精确地控制每幅图形的大小，可以有选择地在`layout()`函数中使用`widths=`和`heights=`两个参数。其形式为：

`widths =` 各列宽度值组成的一个向量
`heights =` 各行高度值组成的一个向量

相对宽度可以直接通过数值指定，绝对宽度（以厘米为单位）可以通过函数`lcm()`来指定。

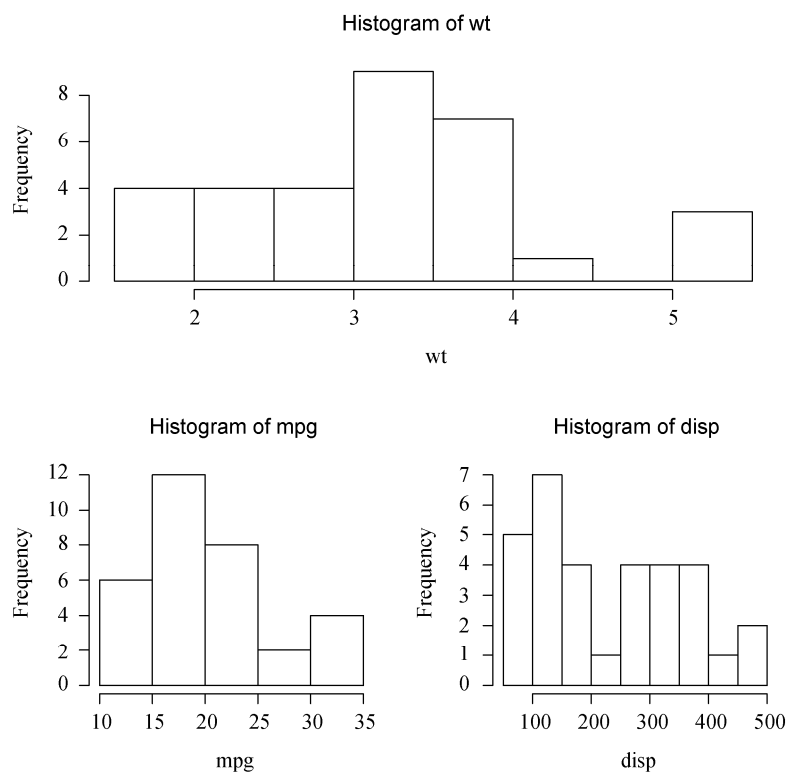


图3-16 使用函数layout()组合的三幅图形，各列宽度为默认值

在以下代码中，我们再次将一幅图形置于第1行，两幅图形置于第2行。但第1行中图形的高度是第2行中图形高度的三分之一。除此之外，右下角图形的宽度是左下角图形宽度的四分之一：

```
attach(mtcars)
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE),
        widths=c(3, 1), heights=c(1, 2))
hist(wt)
hist(mpg)
hist(dis)
detach(mtcars)
```

所得图形如图3-17所示。

如你所见，layout()函数能够让我们轻松地控制最终图形中的子图数量和摆放方式，以及这些子图的相对大小。请参考[help\(layout\)](#)以了解更多细节。

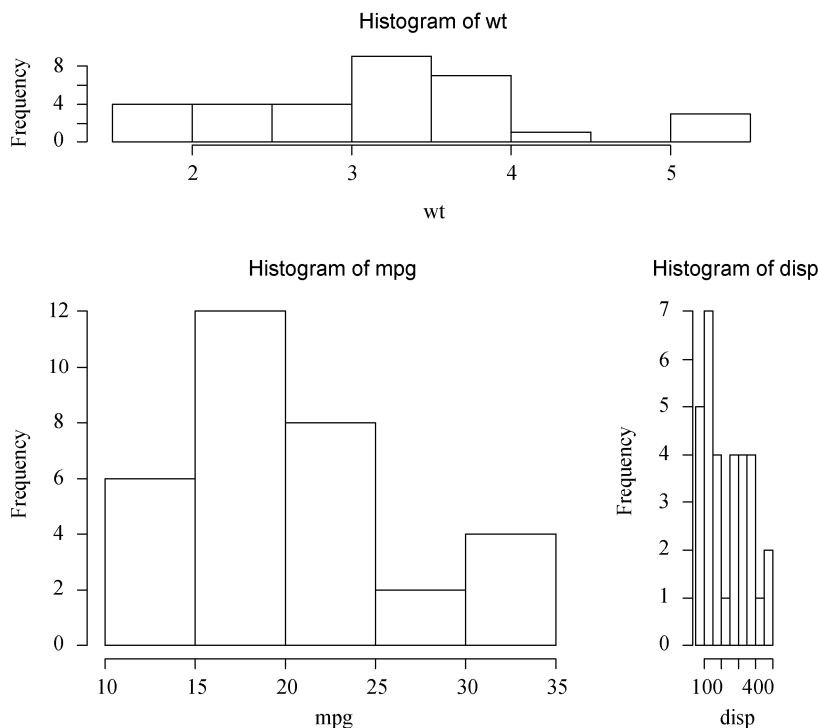


图3-17 使用函数layout()组合的三幅图形，各列宽度为指定值

图形布局的精细控制

可能有很多时候，你想通过排布或叠加若干图形来创建单幅的、有意义的图形，这需要对图形布局的精细控制能力。你可以使用图形参数fig完成这个任务。代码清单3-4通过在散点图上添加两幅箱线图，创建了单幅的增强型图形。结果如图3-18所示。

代码清单3-4 多幅图形布局的精细控制

```
opar <- par(no.readonly=TRUE)
par(fig=c(0, 0.8, 0, 0.8))
plot(mtcars$wt, mtcars$mpg,
     xlab="Miles Per Gallon",
     ylab="Car Weight")

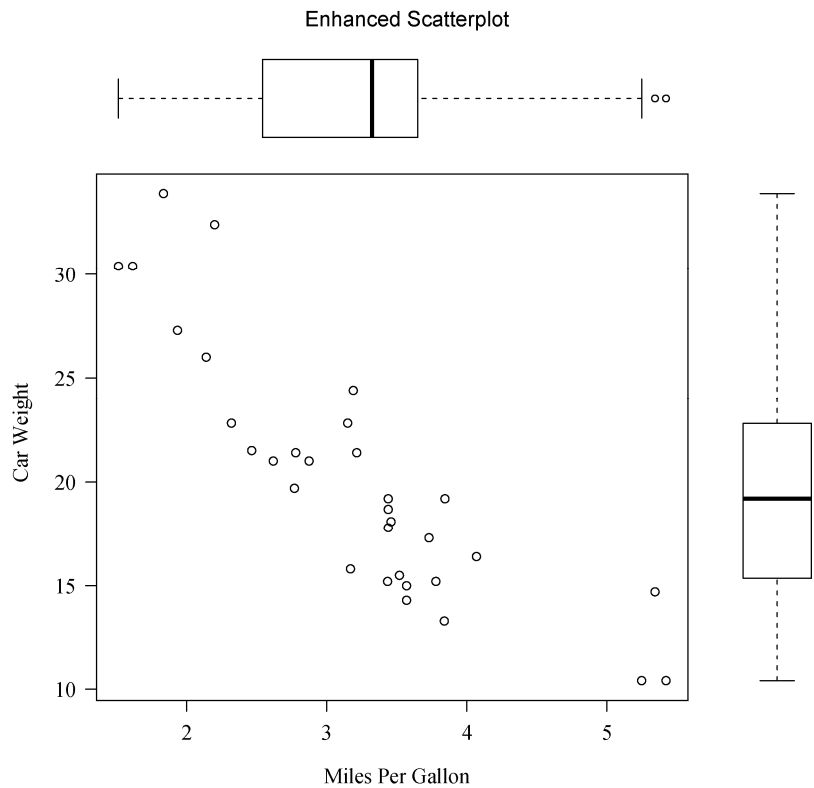
par(fig=c(0, 0.8, 0.55, 1), new=TRUE)
boxplot(mtcars$wt, horizontal=TRUE, axes=FALSE)
par(fig=c(0.65, 1, 0, 0.8), new=TRUE)
boxplot(mtcars$mpg, axes=FALSE)

mtext("Enhanced Scatterplot", side=3, outer=TRUE, line=-3)
par(opar)
```

← 设置散点图

← 在上方添加箱线图

← 在右侧添加箱线图



3

图3-18 边界上添加了两幅箱线图的散点图

要理解这幅图的绘制原理，请试想完整的绘图区域：左下角坐标为(0,0)，而右上角坐标为(1,1)。图3-19是一幅示意图。参数fig=的取值是一个形如c(x1, x2, y1, y2)的数值向量。

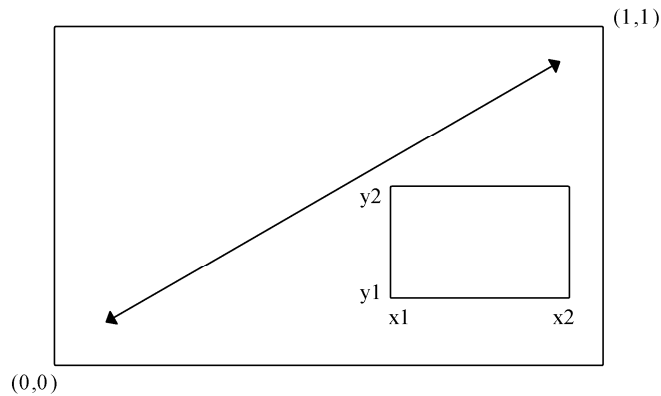


图3-19 使用图形参数fig=指定位置

第一个fig=将散点图设定为占据横向范围0~0.8，纵向范围0~0.8。上方的箱线图横向占据

0~0.8，纵向0.55~1。右侧的箱线图横向占据0.65~1，纵向0~0.8。`fig`=默认会新建一幅图形，所以在添加一幅图到一幅现有图形上时，请设定参数`new=TRUE`。

我将参数选择为0.55而不是0.8，这样上方的图形就不会和散点图拉得太远。类似地，我选择了参数0.65以拉近右侧箱线图和散点图的距离。你需要不断尝试找到合适的位置参数。

注意 各独立子图所需空间的大小可能与设备相关。如果你遇到了“Error in plot.new(): figure margins too large”这样的错误，请尝试在整个图形的范围内修改各个子图占据的区域位置和大小。

你可以使用图形参数`fig`=将若干图形以任意排布方式组合到单幅图形中。稍加练习，你可以通过这种方法极其灵活地创建复杂的视觉呈现。

3.6 小结

本章中，我们回顾了创建图形和以各种格式保存图形的方法。本章的主体则是关于如何修改R绘制的默认图形，以得到更加有用或更吸引人的图形。你学习了如何修改一幅图形的坐标轴、字体、绘图符号、线条和颜色，以及如何添加标题、副标题、标签、文本、图例和参考线，看到了如何指定图形和边界的大小，以及将多幅图形组合为实用的单幅图形。

本章的焦点是那些可以应用于所有图形的通用方法（第16章的lattice图形是一个例外）。后续各章将着眼于特定的图形类型。例如，第7章介绍了对单变量绘图的各种方法；对变量间关系绘图的方法将于第11章讨论；在第16章中，我们则讨论高级的绘图方法，包括lattice图形（可以分水平展示变量间的关系）和交互式图形。交互式图形能让你使用鼠标动态探索数据中的关系。

在其他各章中，我们将会讨论对于某些统计方法来说特别实用的数据可视化方法。图形是现代数据分析的核心组成部分，所以我将尽力将它们整合到各类统计方法的讨论中。

在前一章中，我们讨论了一系列输入或导入数据到R中的方法。遗憾的是，现实数据极少以直接可用的格式出现。下一章，我们将关注如何将数据转换或修改为更有助于分析的形式。