

体、缩小为默认大小的75%。之后，我们使用红色实心圆圈和虚线创建了第一幅图形，并使用绿色填充的绿色菱形加蓝色边框和蓝色虚线创建了第二幅图形。最后，我们还还原了初始的图形参数设置。

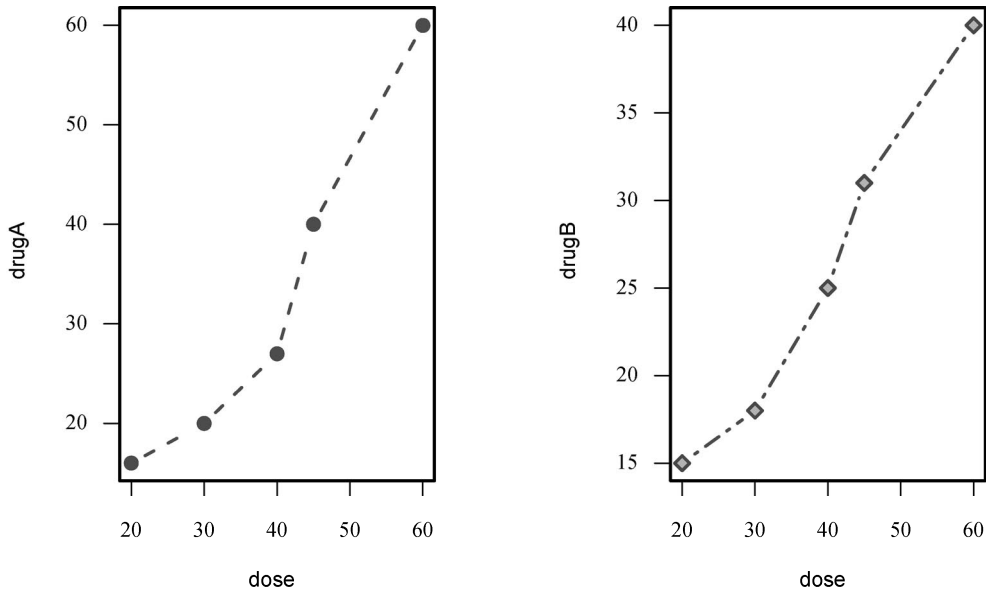


图3-7 药物A和药物B剂量与响应的折线图（另见彩插图3-7）

值得注意的是，通过`par()`设定的参数对两幅图都有效，而在绘图函数中指定的参数仅对那个特定图形有效。观察图3-7可以发现，图形的呈现上还有一点缺陷。这两幅图都缺少标题，并且纵轴的刻度单位不同，这无疑限制了我们直接比较两种药物的能力。同时，坐标轴的标签（名称）也应当提供更多的信息。

下一节中，我们将转而探讨如何自定义文本标注（如标题和标签）和坐标轴。要了解可用图形参数的更多信息，请参阅`help(par)`。

3.4 添加文本、自定义坐标轴和图例

除了图形参数，许多高级绘图函数（例如`plot`、`hist`、`boxplot`）也允许自行设定坐标轴和文本标注选项。举例来说，以下代码在图形上添加了标题（`main`）、副标题（`sub`）、坐标轴标签（`xlab`、`ylab`）并指定了坐标轴范围（`xlim`、`ylim`）。结果如图3-8所示。

```
plot(dose, drugA, type="b",
     col="red", lty=2, pch=2, lwd=2,
     main="Clinical Trials for Drug A",
     sub="This is hypothetical data",
     xlab="Dosage", ylab="Drug Response",
     xlim=c(0, 60), ylim=c(0, 70))
```

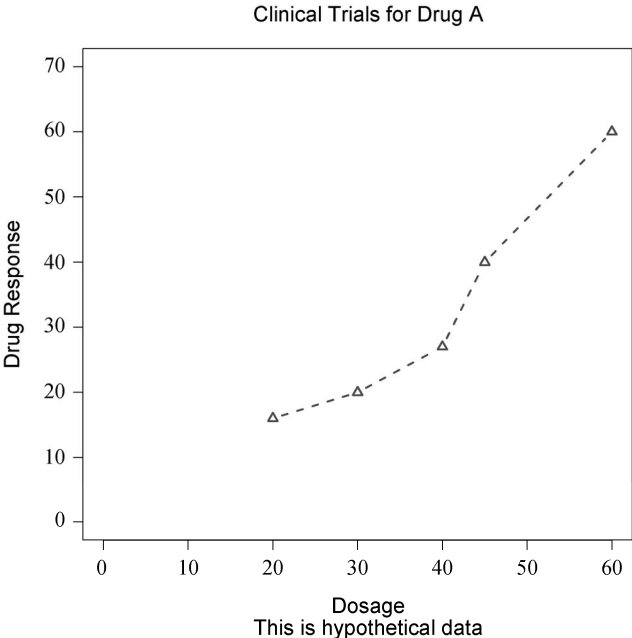


图3-8 药物A剂量和响应的折线图。添加了标题、副标题和自定义的坐标轴

再次提醒，并非所有函数都支持这些选项。请参考相应函数的帮助以了解其可以接受哪些选项。从更精细的控制和模块化的角度考虑，你可以使用本节余下部分描述的函数来控制标题、坐标轴、图例和文本标注的外观。

注意 某些高级绘图函数已经包含了默认的标题和标签。你可以通过在`plot()`语句或单独的`par()`语句中添加`ann=FALSE`来移除它们。

3.4.1 标题

可以使用`title()`函数为图形添加标题和坐标轴标签。调用格式为：

```
title(main="main title", sub="sub-title",
      xlab="x-axis label", ylab="y-axis label")
```

函数`title()`中亦可指定其他图形参数（如文本大小、字体、旋转角度和颜色）。举例来说，以下代码将生成红色的标题和蓝色的副标题，以及较默认大小小25%的绿色x轴、y轴标签：

```
title(main="My Title", col.main="red",
      sub="My Sub-title", col.sub="blue",
      xlab="My X label", ylab="My Y label",
      col.lab="green", cex.lab=0.75)
```

3.4.2 坐标轴

你可以使用函数axis()来创建自定义的坐标轴，而非使用R中的默认坐标轴。其格式为：

```
axis(side, at=, labels=, pos=, lty=, col=, las=, tck=, ...)
```

各参数已详述于表3-7中。

表3-7 坐标轴选项

选 项	描 述
side	一个整数，表示在图形的哪边绘制坐标轴（1=下，2=左，3=上，4=右）
at	一个数值型向量，表示需要绘制刻度线的位置
labels	一个字符型向量，表示置于刻度线旁边的文字标签（如果为NULL，则将直接使用at中的值）
pos	坐标轴线绘制位置的坐标（即与另一条坐标轴相交位置的值）
lty	线条类型
col	线条和刻度线颜色
las	标签是否平行于（=0）或垂直于（=2）坐标轴
tck	刻度线的长度，以相对于绘图区域大小的分数表示（负值表示在图形外侧，正值表示在图形内侧，0表示禁用刻度，1表示绘制网格线）；默认值为-0.01
(...)	其他图形参数

创建自定义坐标轴时，你应当禁用高级绘图函数自动生成的坐标轴。参数axes=FALSE将禁用全部坐标轴（包括坐标轴框架线，除非你添加了参数frame.plot=TRUE）。参数xaxt="n"和 yaxt="n"将分别禁用X轴或Y轴（会留下框架线，只是去除了刻度）。代码清单3-2中是一个稍显笨拙和夸张的例子，它演示了我们到目前为止讨论过的各种图形特征。结果如图3-9所示。

代码清单3-2 自定义坐标轴的示例

```
x <- c(1:10)                                <— 生成数据
y <- x
z <- 10/x

opar <- par(no.readonly=TRUE)

par(mar=c(5, 4, 4, 8) + 0.1)                <— 增加边界大小

plot(x, y, type="b",                          <— 绘制x对y的图形
     pch=21, col="red",
     yaxt="n", lty=3, ann=FALSE)

lines(x, z, type="b", pch=22, col="blue", lty=2)    <— 添加x对1/x的直线

axis(2, at=x, labels=x, col.axis="red", las=2)      <— 绘制你自己的坐标轴

axis(4, at=z, labels=round(z, digits=2),
     col.axis="blue", las=2, cex.axis=0.7, tck=-.01)
```

```

mtext("y=1/x", side=4, line=3, cex.lab=1, las=2, col="blue")

title("An Example of Creative Axes",
      xlab="X values",
      ylab="Y=X")

par(opar)

```

添加标题和文本

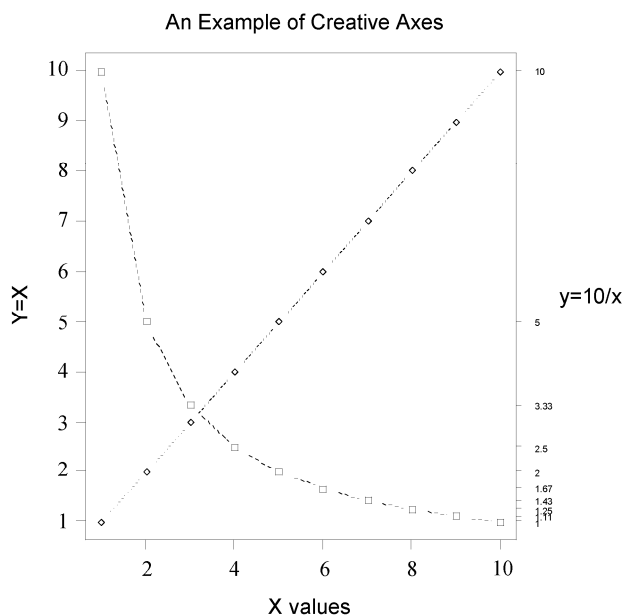


图3-9 各种坐标轴选项的演示

到目前为止，我们已经讨论过代码清单3-2中除`lines()`和`mtext()`以外的所有函数。使用`plot()`语句可以新建一幅图形。而使用`lines()`语句，你可以为一幅现有图形添加新的图形元素。在3.4.4节中，你会再次用到它，在同一幅图中绘制药物A和药物B的响应情况。函数`mtext()`用于在图形的边界添加文本。我们将在3.4.5节中讲到函数`mtext()`，同时会在第11章中更充分地讨论`lines()`函数。

次要刻度线

注意，我们最近创建的图形都只拥有主刻度线，却没有次要刻度线。要创建次要刻度线，你需要使用Hmisc包中的`minor.tick()`函数。如果你尚未安装Hmisc包，请先安装它（参考1.4.2节）。你可以使用代码：

```

library(Hmisc)
minor.tick(nx=n, ny=n, tick.ratio=n)

```

来添加次要刻度线。其中`nx`和`ny`分别指定了X轴和Y轴每两条主刻度线之间通过次要刻度线划分得到的区间个数。`tick.ratio`表示次要刻度线相对于主刻度线的大小比例。当前的主刻度线长度可以使用`par("tck")`获取。举例来说，下列语句将在X轴的每两条主刻度线之间添加1条次要

刻度线，并在Y轴的每两条主刻度线之间添加2条次要刻度线：

```
minor.tick(nx=2, ny=3, tick.ratio=0.5)
```

次要刻度线的长度将是主刻度线的一半。3.4.4节中给出了添加次要刻度线的一个例子（代码清单3-3和图3-10）。

3.4.3 参考线

函数`abline()`可以用来为图形添加参考线。其使用格式为：

```
abline(h=yvalues, v=xvalues)
```

函数`abline()`中也可以指定其他图形参数（如线条类型、颜色和宽度）。举例来说：

```
abline(h=c(1,5,7))
```

在y为1、5、7的位置添加了水平实线，而代码：

```
abline(v=seq(1, 10, 2), lty=2, col="blue")
```

则在x为1、3、5、7、9的位置添加了垂直的蓝色虚线。代码清单3-3为我们的药物效果图在y=30的位置创建了一条参考线。结果如图3-10所示。

3.4.4 图例

当图形中包含的数据不止一组时，图例可以帮助你辨别出每个条形、扇形区域或折线各代表哪一类数据。我们可以使用函数`legend()`来添加图例（果然不出所料）。其使用格式为：

```
legend(location, title, legend, ...)
```

常用选项详述于表3-8中。

表3-8 图例选项

选 项	描 述
location	有许多方式可以指定图例的位置。你可以直接给定图例左上角的x、y坐标，也可以执行 <code>locator(1)</code> ，然后通过鼠标单击给出图例的位置，还可以使用关键字 <code>bottom</code> 、 <code>bottomleft</code> 、 <code>left</code> 、 <code>topleft</code> 、 <code>top</code> 、 <code>topright</code> 、 <code>right</code> 、 <code>bottomright</code> 或 <code>center</code> 放置图例。如果你使用了以上某个关键字，那么可以同时使用参数 <code>inset</code> =指定图例向图形内侧移动的大小（以绘图区域大小的分数表示）
title	图例标题的字符串（可选）
legend	图例标签组成的字符型向量
...	其他选项。如果图例标示的是颜色不同的线条，需要指定 <code>col</code> =加上颜色值组成的向量。如果图例标示的是符号不同的点，则需指定 <code>pch</code> =加上符号的代码组成的向量。如果图例标示的是不同的线条宽度或线条类型，请使用 <code>lwd</code> =或 <code>lty</code> =加上宽度值或类型值组成的向量。要为图例创建颜色填充的盒形（常见于条形图、箱线图或饼图），需要使用参数 <code>fill</code> =加上颜色值组成的向量

其他常用的图例选项包括用于指定盒子样式的`bty`、指定背景色的`bg`、指定大小的`cex`，以及指定文本颜色的`text.col`。指定`horiz=TRUE`将会水平放置图例，而不是垂直放置。关于图

例的更多细节，请参考`help(legend)`。这份帮助中给出的示例都特别有用。

让我们看看对药物数据作图的一个例子（代码清单3-3）。你将再次使用我们目前为止讲到的许多图形功能。结果如图3-10所示。

代码清单3-3 依剂量对比药物A和药物B的响应情况

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)

opar <- par(no.readonly=TRUE)

par(lwd=2, cex=1.5, font.lab=2)

plot(dose, drugA, type="b",
      pch=15, lty=1, col="red", ylim=c(0, 60),
      main="Drug A vs. Drug B",
      xlab="Drug Dosage", ylab="Drug Response")

lines(dose, drugB, type="b",
      pch=17, lty=2, col="blue")

abline(h=c(30), lwd=1.5, lty=2, col="gray")

library(Hmisc)
minor.tick(nx=3, ny=3, tick.ratio=0.5)

legend("topleft", inset=.05, title="Drug Type", c("A","B"),
      lty=c(1, 2), pch=c(15, 17), col=c("red", "blue"))

par(opar)
```

增加线条、文本、符号、
标签的宽度或大小

绘制图形

添加次要刻度线

添加图例

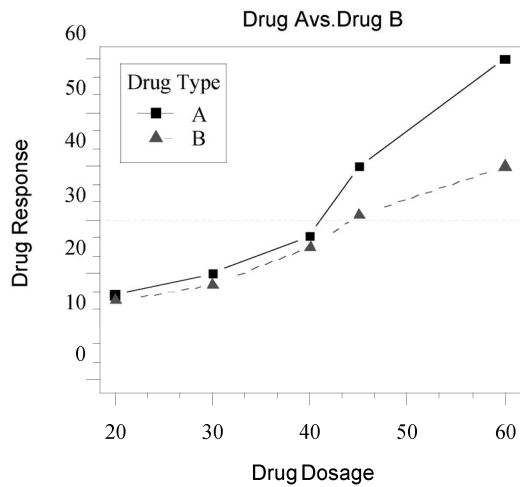


图3-10 进行标注后的图形，对比了药物A和药物B的效果

图3-10的几乎所有外观元素都可以使用本章中讨论过的选项进行修改。除此之外，还有很多其他方式可以指定想要的选项。最后一种需要研究的图形标注是向图形本身添加文本，请继续读下一节。

3.4.5 文本标注

我们可以通过函数`text()`和`mtext()`将文本添加到图形上。`text()`可向绘图区域内部添加文本，而`mtext()`则向图形的四个边界之一添加文本。使用格式分别为：

```
text(location, "text to place", pos, ...)
mtext("text to place", side, line=n, ...)
```

常用选项列于表3-9中。

表3-9 函数`text()`和`mtext()`的选项

选 项	描 述
location	文本的位置参数。可为一对x,y坐标，也可通过指定location为locator(1)使用鼠标交互式地确定摆放位置
pos	文本相对于位置参数的方位。1=下, 2=左, 3=上, 4=右。如果指定了pos, 就可以同时指定参数offset=作为偏移量，以相对于单个字符宽度的比例表示
side	指定用来放置文本的边。1=下, 2=左, 3=上, 4=右。你可以指定参数line=来内移或外移文本，随着值的增加，文本将外移。也可使用adj=0将文本向左下对齐，或使用adj=1右上对齐

其他常用的选项有`cex`、`col`和`font`（分别用来调整字号、颜色和字体样式）。

除了用来添加文本标注以外，`text()`函数也通常用来标示图形中的点。我们只需指定一系列的x、y坐标作为位置参数，同时以向量的形式指定要放置的文本。x、y和文本标签向量的长度应当相同。下面给出了一个示例，结果如图3-11所示。

```
attach(mtcars)
plot(wt, mpg,
     main="Mileage vs. Car Weight",
     xlab="Weight", ylab="Mileage",
     pch=18, col="blue")
text(wt, mpg,
     row.names(mtcars),
     cex=0.6, pos=4, col="red")
detach(mtcars)
```

这里，我们针对数据框`mtcars`提供的32种车型的车重和每加仑汽油行驶英里数绘制了散点图。函数`text()`被用来在各个数据点右侧添加车辆型号。各点的标签大小被缩小了40%，颜色为红色。

作为第二个示例，以下是一段展示不同字体族的代码：

```
opar <- par(no.readonly=TRUE)
par(cex=1.5)
plot(1:7, 1:7, type="n")
text(3, 3, "Example of default text")
text(4, 4, family="mono", "Example of mono-spaced text")
text(5, 5, family="serif", "Example of serif text")
par(opar)
```

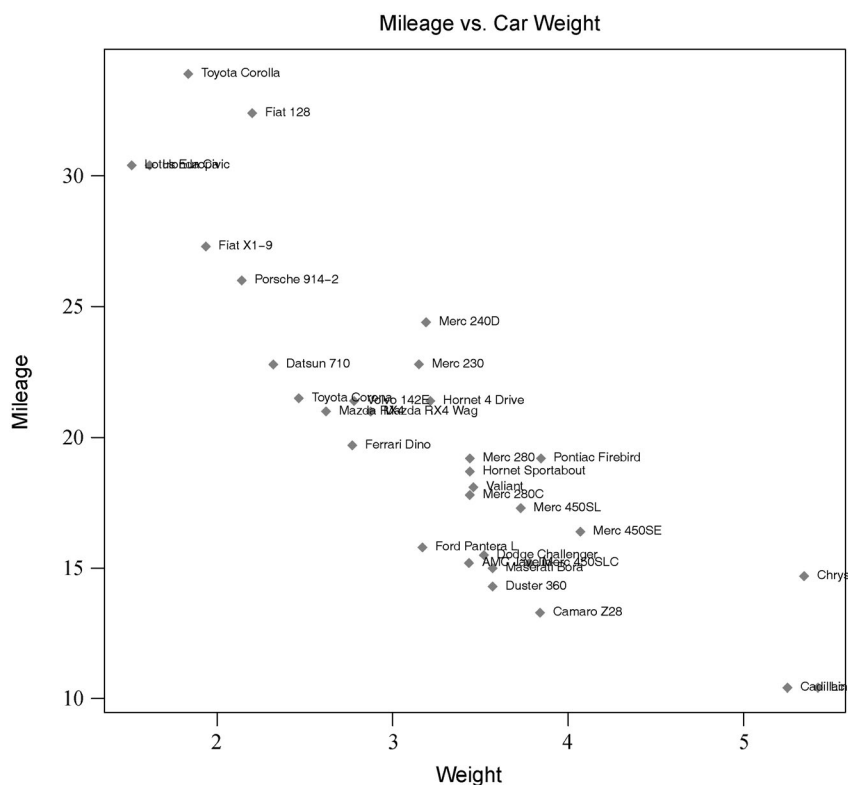


图3-11 一幅散点图（车重与每加仑汽油行驶英里数）的示例，各点均添加了标签（车型）

在Windows系统中输出的结果如图3-12所示。这里为了获得更好的显示效果，我们使用`par()`函数增大了字号。

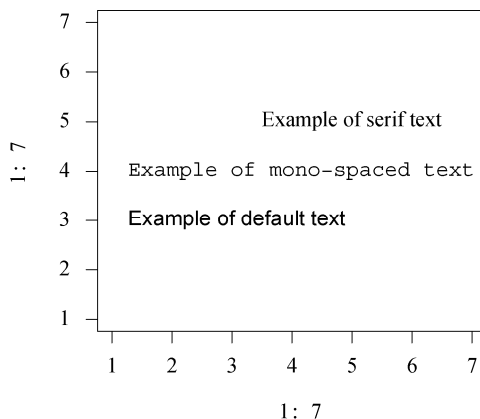


图3-12 Windows中不同字体族的示例

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	
<code>x + y</code>	$x + y$	<code>sqrt(x)</code>	\sqrt{x}
<code>x - y</code>	$x - y$	<code>sqrt(x, y)</code>	$\sqrt[x]{y}$
<code>x * y</code>	xy	Relations	
<code>x / y</code>	x / y	<code>x == y</code>	$x = y$
<code>x %+-% y</code>	$x \pm y$	<code>x != y</code>	$x \uparrow y$
<code>x %/% y</code>	$x \sqrt{y}$	<code>x < y</code>	$x < y$
<code>x %*% y</code>	$x \times y$	<code>x <= y</code>	$x \leq y$
<code>x %.% y</code>	$x \cdot y$	<code>x > y</code>	$x > y$
<code>-x</code>	$-x$	<code>x >= y</code>	$x \geq y$
<code>+x</code>	$+x$	<code>x %~% y</code>	$x \oplus y$
Sub/Superscripts		<code>x %~% y</code>	$x \equiv y$
<code>x[i]</code>	x_i	<code>x %==% y</code>	$x \equiv y$
<code>x^2</code>	x^2	<code>x %prop% y</code>	$x \propto y$
Juxtaposition		Typeface	
<code>x * y</code>	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。