角线很远)。当变量数众多，变量间的相关性变化很大时，该方法特别有用。在第16章，你还将会看到其他散点图矩阵的例子。

11.1.2　高密度散点图

当数据点重叠很严重时，用散点图来观察变量关系就显得“力不从心”了。下面是一个人为设计的例子，其中10 000个观测点分布在两个重叠的数据群中：

```
set.seed(1234)

n <- 10000
c1 <- matrix(rnorm(n, mean=0, sd=.5), ncol=2)
c2 <- matrix(rnorm(n, mean=3, sd=2), ncol=2)
mydata <- rbind(c1, c2)
mydata <- as.data.frame(mydata)
names(mydata) <- c("x", "y")
```

若用下面的代码生成一幅标准的散点图：

```
with(mydata,
     plot(x, y, pch=19, main="Scatter Plot with 10,000 Observations"))
```

你将会得到如图11-7所示的图形。

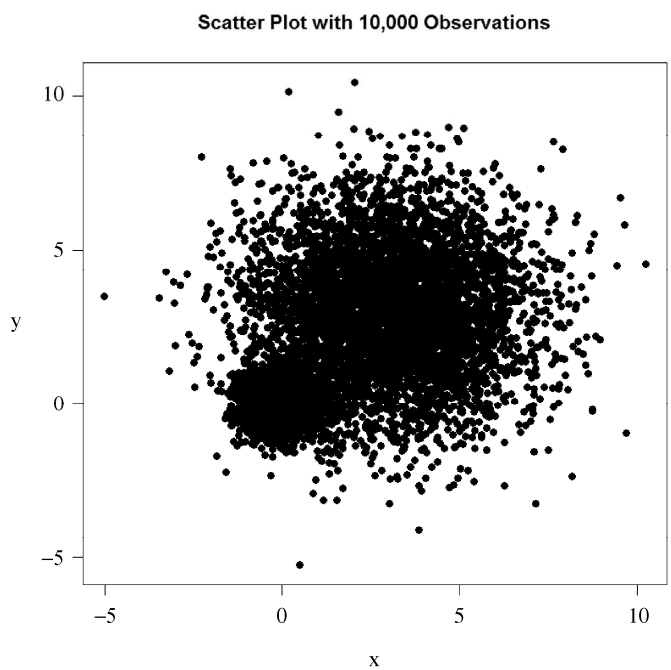

图11-7　10 000个观测点的散点图，严重的重叠导致很难识别哪里数据点的密度最大

图11-7中，数据点的重叠导致识别x与y间的关系变得异常困难。针对这种情况，R提供了一

些解决办法。你可以使用封箱、颜色和透明度来指明图中任意点上重叠点的数目。

smoothScatter()函数可利用核密度估计生成用颜色密度来表示点分布的散点图。代码如下：

```
with(mydata,
     smoothScatter(x, y, main="Scatterplot Colored by Smoothed Densities"))
```

生成图形见图11-8。

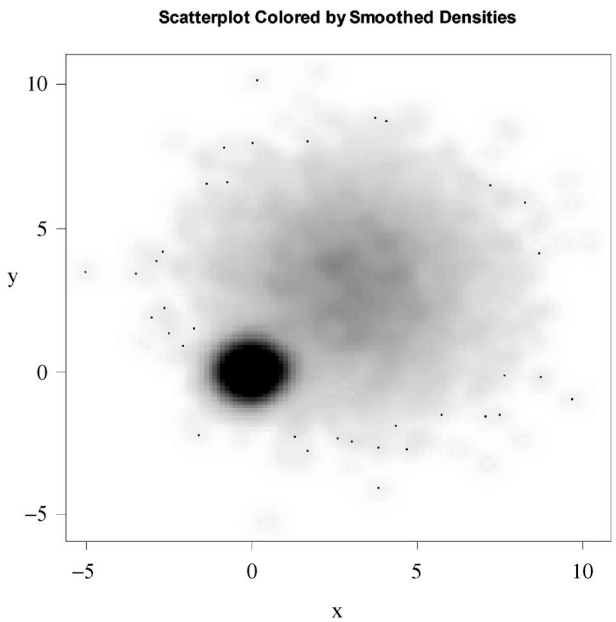

图11-8 smoothScatter()利用光平滑密度估计绘制的散点图。此处密度易读性更强

与上面的方法不同，hexbin包中的hexbin()函数将二元变量的封箱放到六边形单元格中（图形比名称更直观）。示例如下：

```
library(hexbin)
with(mydata, {
     bin <- hexbin(x, y, xbins=50)
     plot(bin, main="Hexagonal Binning with 10,000 Observations")
     })
```

你将得到如图11-9所示的散点图。

最后，IDPmisc包中的iplot()函数也可通过颜色来展示点的密度（在某特定点上数据点的数目）。代码如下：

```
library(IDPmisc)
with(mydata,
     iplot(x, y, main="Image Scatter Plot with Color Indicating Density"))
```

生成图形见图11-10。

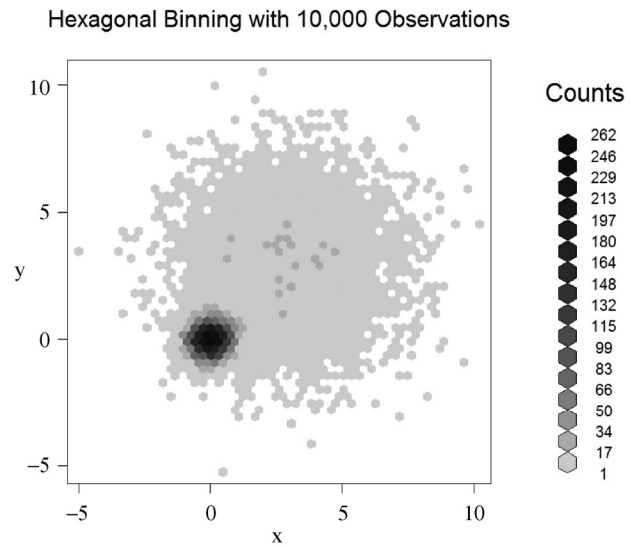

图11-9　用六边形封箱图展示的各点上覆盖观测点数目的散点图。通过图例，数据的集中度很容易计算和观察

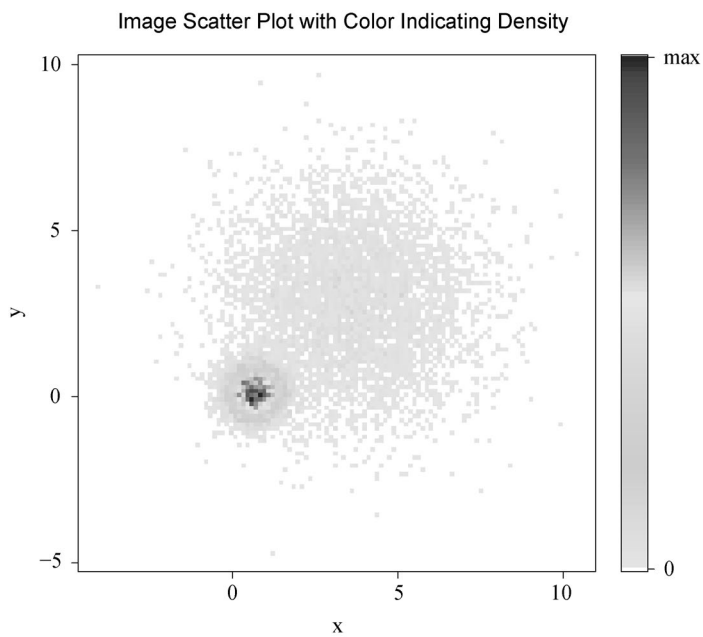

图11-10　含10 000个观测的散点图，其中密度通过颜色标识。数据集中度很容易辨识

综上可见，基础包中的`smoothScatter()`函数，以及`IDPmisc`包中的`ipairs()`函数都可以对大数据集创建可读性较好的散点图矩阵。通过`?smoothScatter`和`?ipairs`可获得更多的示例。