

完整的 ggplot2 教程 - 第 1 部分 | ggplot2 (全 R 码) 简介

Source Credit: <http://r-statistics.co/Complete-Ggplot2-Tutorial-Part1-With-R-Code.html>

之前我们看到了一个简单的使用 ggplot2 软件包制作图表的教程。它很快涉及了制作 ggplot 的各个方面。现在，这是一个完整而完整的教程。我从头开始讨论如何构建和定制几乎任何 ggplot。它涉及使情节更有效和更具视觉吸引力的原则，步骤和细微差别。所以，为了实际的目的，我希望本教程能够很好地作为书签参考，这对您的日常制作有用。这是一个关于 ggplot2 的三部分教程的第一部分，这是一个美学上令人愉快（并且非常流行）的 R 图形框架。本教程主要面向那些具有 R 编程语言基础知识并希望制作复杂漂亮的图表与 R ggplot2。

第 1 部分：ggplot2 简介，涵盖了构建简单 ggplots 和修改组件和美学的知识。

第 2 部分：自定义外观和感觉，是关于更高级的自定义，比如操纵图例，注释，多面体以及自定义布局

第 3 部分：前 50 名 Ggplot2 可视化 - 主列表，将第 1 部分和第 2 部分中学到的内容应用于构建其他类型的 ggplots，如条形图，箱形图等。

1. 了解 Ggplot 语法

如果你是初学者或主要使用基本图形工作，构建 ggplots 的语法可能会令人费解。主要区别在于，与基本图形不同，ggplot 可以处理数据帧而不是单独的向量。所有使绘图所需的数据通常都包含在提供给 ggplot () 本身的数据框中，或者可以提供给各个 geom。稍后更多。

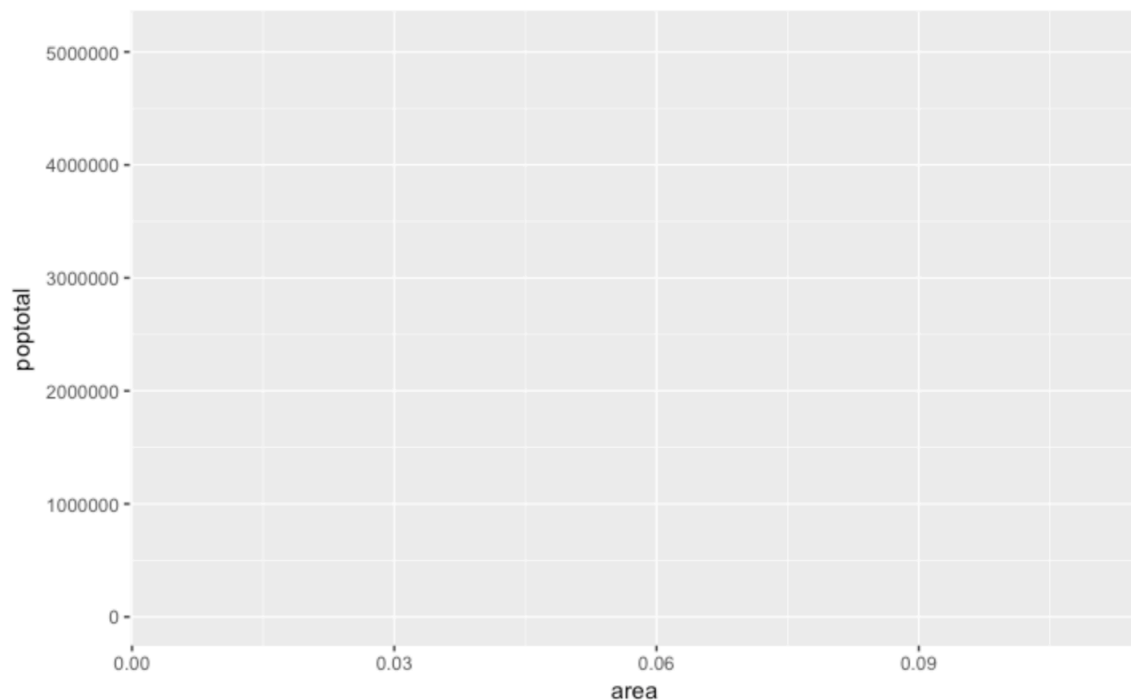
第二个显而易见的特征是，您可以通过向使用 ggplot () 函数创建的现有图添加更多图层（和主题）来继续增强绘图。

让我们初始化一个基于中西部数据集的基本 ggplot。

配置

```
# Setup
options(scipen=999) # 关闭科学记数法，如 1e + 06
library(ggplot2)
data("midwest", package = "ggplot2") # 加载数据
# midwest <- read.csv("http://goo.gl/G1K41K") # alt source
# Init Ggplot
```

```
ggplot(midwest, aes(x=area, y=poptotal)) # area and poptotal are columns in 'midwest'
```



初始化 Ggplot

```
ggplot (midwest, aes (x = area, y = poptotal) ) # area 和 poptotal 是'midwest'中的列
```

绘制了空白 ggplot。即使指定了 x 和 y，它里面也没有任何点或线。这是因为，ggplot 并不假定你的意思是绘制散点图或折线图。我只告诉 ggplot 要使用的数据集以及 X 和 Y 轴应该使用哪些列。我没有明确要求它提出任何观点。

另请注意，aes () 函数用于指定 X 轴和 Y 轴。这是因为，必须在 aes () 函数内指定属于源数据框一部分的任何信息。

2.如何制作一个简单的散点图

让我们通过使用称为 geom_point 的几何图层添加点来在空白 ggplot 顶部绘制散点图。

```
library(ggplot2)  
ggplot(midwest, aes(x=area, y=poptotal)) + geom_point()
```

我们得到了一个基本的散点图，每个点代表一个县。但是，它缺少一些基本组成部分，如情节标题，有意义的轴标等。此外，大多数要点都集中在情节的底部，这并不是很好。你会看到如何在即将到来的步骤中纠正这些问题。

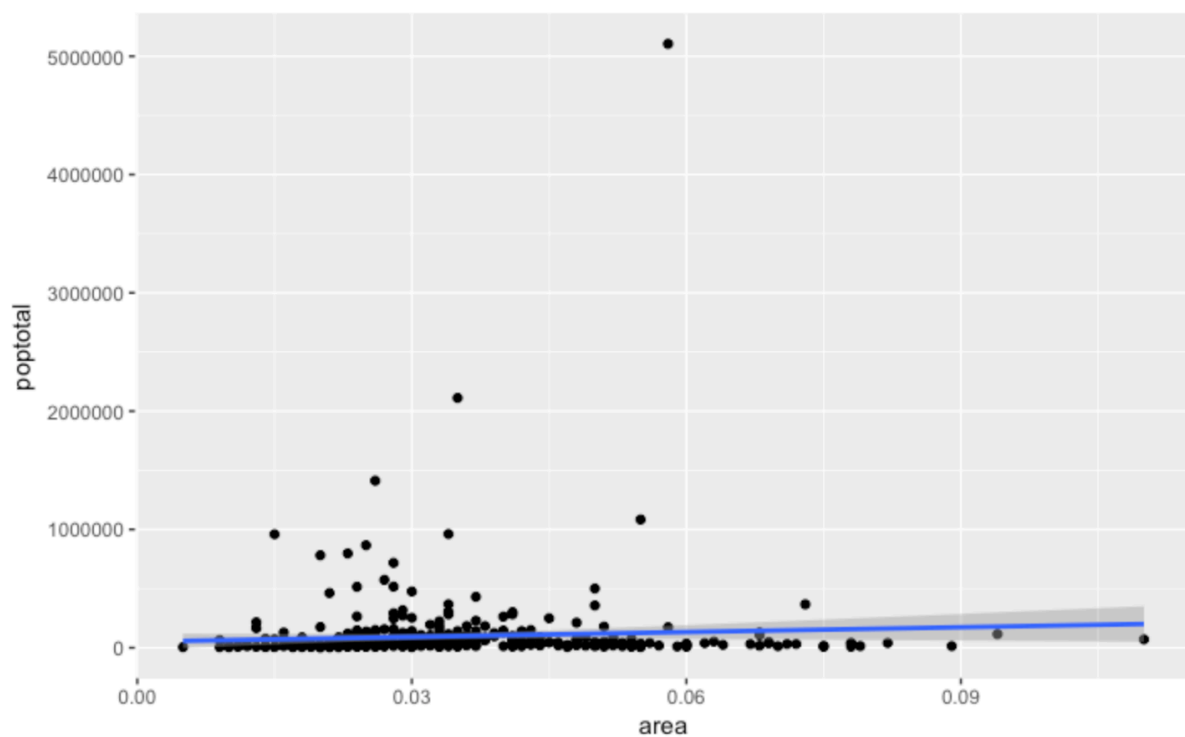
像 `geom_point()` 一样，我们将在本系列教程的后续部分中看到许多这样的 `geom` 层。现在，让我们使用 `geom_smooth(method='lm')` 添加一个平滑层。由于该方法设置为 `lm`（线性模型的缩写），因此绘制出最适合的线。

库 (GGPLOT2)

```
library(ggplot2)

g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm") # set se
=FALSE to turnoff confidence bands

plot(g)
```



最适合的线条是蓝色的。你能找出 `geom_smooth` 有什么其他方法可用吗？（注意：请参阅 `?geom_smooth`）。您可能已经注意到，大多数点位于图表的底部，看起来并不太好。所以，让我们改变 Y 轴的限制，把重点放在下半部分。

3.调整 X 和 Y 轴限制

X 和 Y 轴限制可以通过两种方式进行控制。

方法 1：删除范围外的点

与原始数据相比，这将改变最佳拟合线或平滑线。

这可以通过 `xlim()` 和 `ylim()` 来完成。您可以传递长度为 2 的数值向量（最大值和最小值），或者只传递最大值和最小值。

```
library(ggplot2)

g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm") # set se
=FALSE to turnoff confidence bands

# Delete the points outside the limits

g + xlim(c(0, 0.1)) + ylim(c(0, 1000000)) # deletes points

# g + xlim(0, 0.1) + ylim(0, 1000000) # deletes points
```

在这种情况下，图表不是从头开始构建的，而是建立在 `g` 之上的。这是因为，前面的绘图存储为 `g`，一个 `ggplot` 对象，它在被调用时会重现原始绘图。使用 `ggplot`，您可以在此图上添加更多图层，主题和其他设置。

您是否注意到与最初的情节相比，最合适的线变得更加水平？这是因为，当使用 `xlim()` 和 `ylim()` 时，指定范围以外的点将被删除，并且在绘制最佳拟合线时使用 `geom_smooth(method='lm')` 将不会被考虑。当您想知道当某些极值（或异常值）被移除时，最佳拟合线如何改变，此功能可能派上用场。

方法 2：放大

另一种方法是通过放大感兴趣区域而不删除点来更改 X 和 Y 轴限制。这是使用 `coord_cartesian()` 完成的。

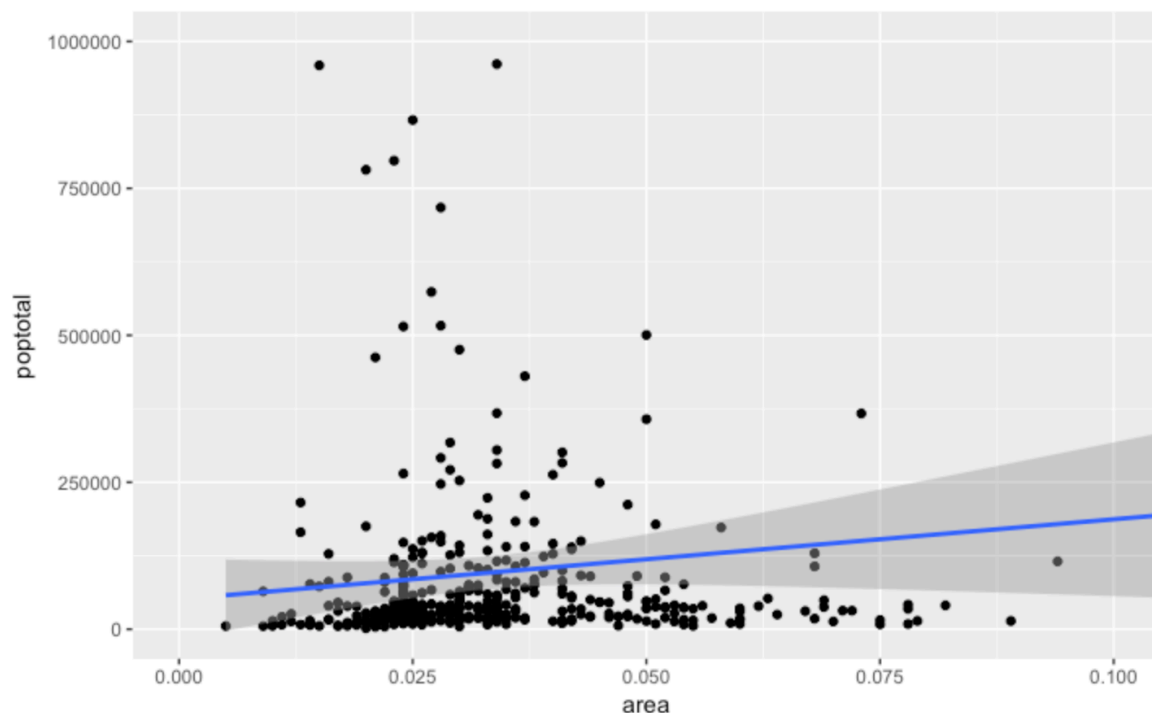
我们将这个图存储为 `g1`。

```
library(ggplot2)

g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm") # set se
=FALSE to turnoff confidence bands

# Zoom in without deleting the points outside the limits.
# As a result, the line of best fit is the same as the original plot.

g1 <- g + coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) # zooms in
plot(g1)
```



由于考虑了所有的点，所以最适合的线路没有改变。

4. 如何更改标题和轴标签

我已经将这个存储为 `g1`。我们添加 X 和 Y 轴的标题和标签。这可以通过带有标题，`x` 和 `y` 参数的 `labs()` 函数一次完成。另一种选择是使用 `ggtitle()`，`xlab()` 和 `ylab()`。

```
library(ggplot2)
```

```

g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm") # set se
=FALSE to turnoff confidence bands

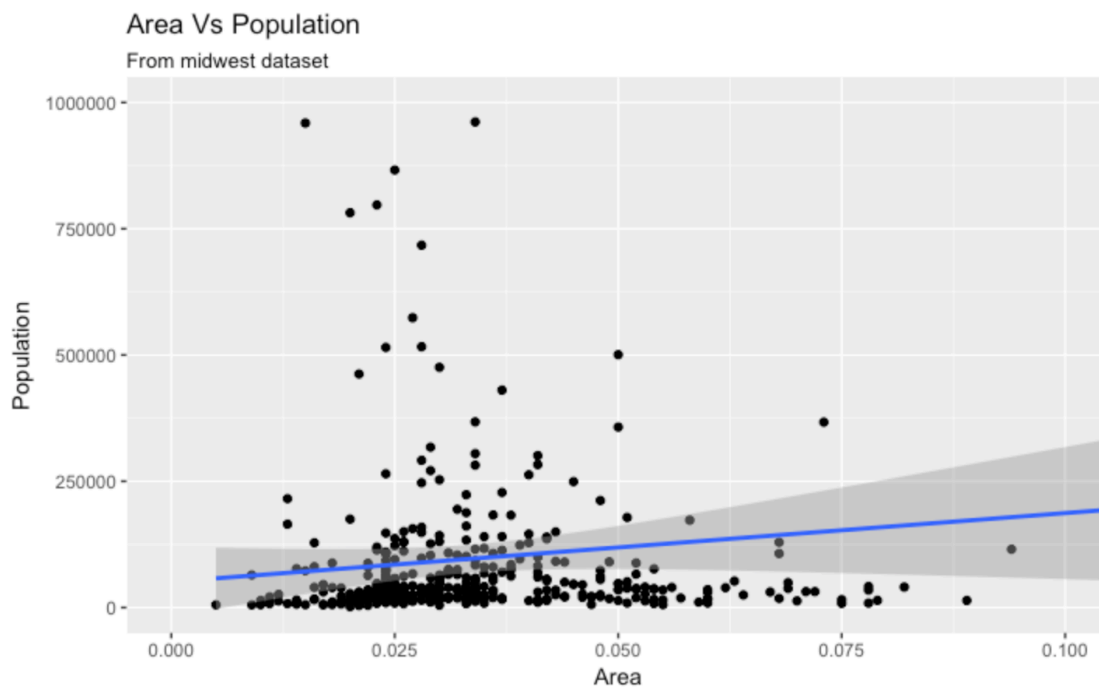
g1 <- g + coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) # zooms in

# Add Title and Labels
g1 + labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Area", caption=
"Midwest Demographics")

# or

g1 + ggtitle("Area Vs Population", subtitle="From midwest dataset") + xlab("Area") + ylab("Population")

```



优秀！所以这里是完整的函数调用。

```

# Full Plot call
library(ggplot2)
ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point() +
  geom_smooth(method="lm") +
  coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +

```

```
labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Area", caption="Midwest Demographics")
```

5.如何更改点的颜色和大小

如何将颜色和大小更改为静态？

我们可以通过修改相应的 `geoms` 来改变 `geom` 层的美学。让我们将点和线的颜色更改为静态值。

```
library(ggplot2)

gg <- ggplot(midwest, aes(x=area, y=poptotal)) +

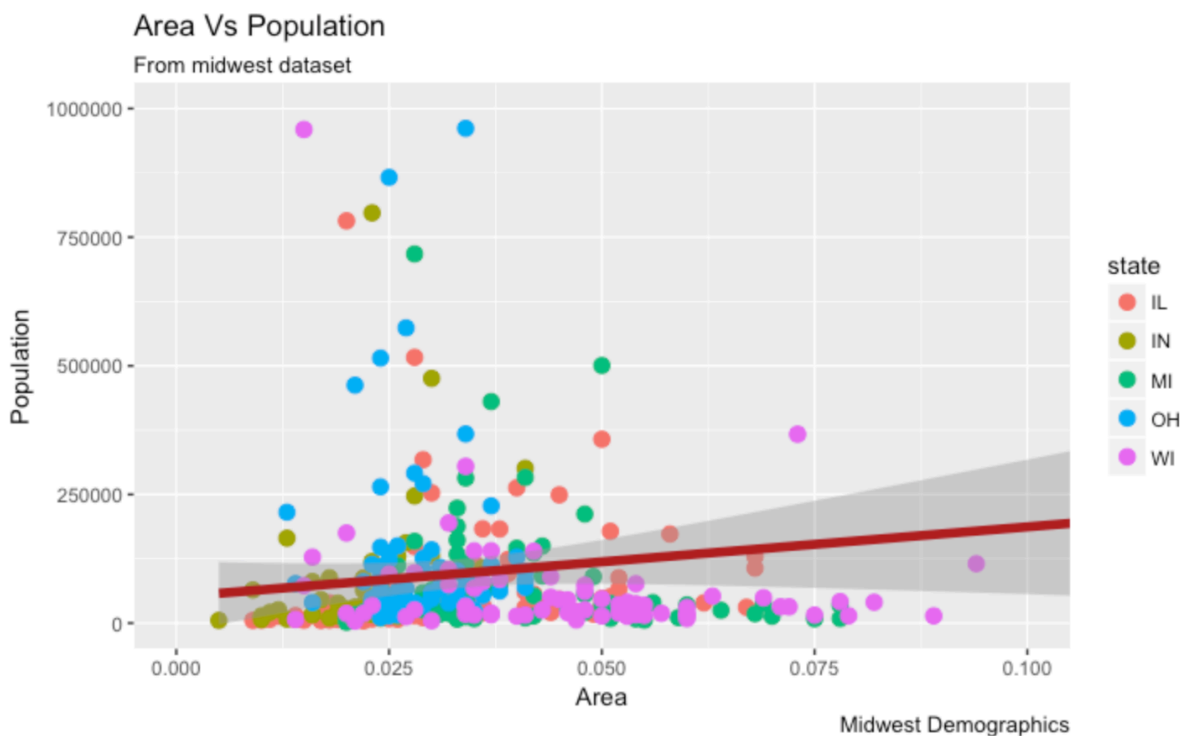
  geom_point(aes(col=state), size=3) + # Set color to vary based on state categories.

  geom_smooth(method="lm", col="firebrick", size=2) +

  coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +

  labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Area", caption="Midwest Demographics")

plot(gg)
```



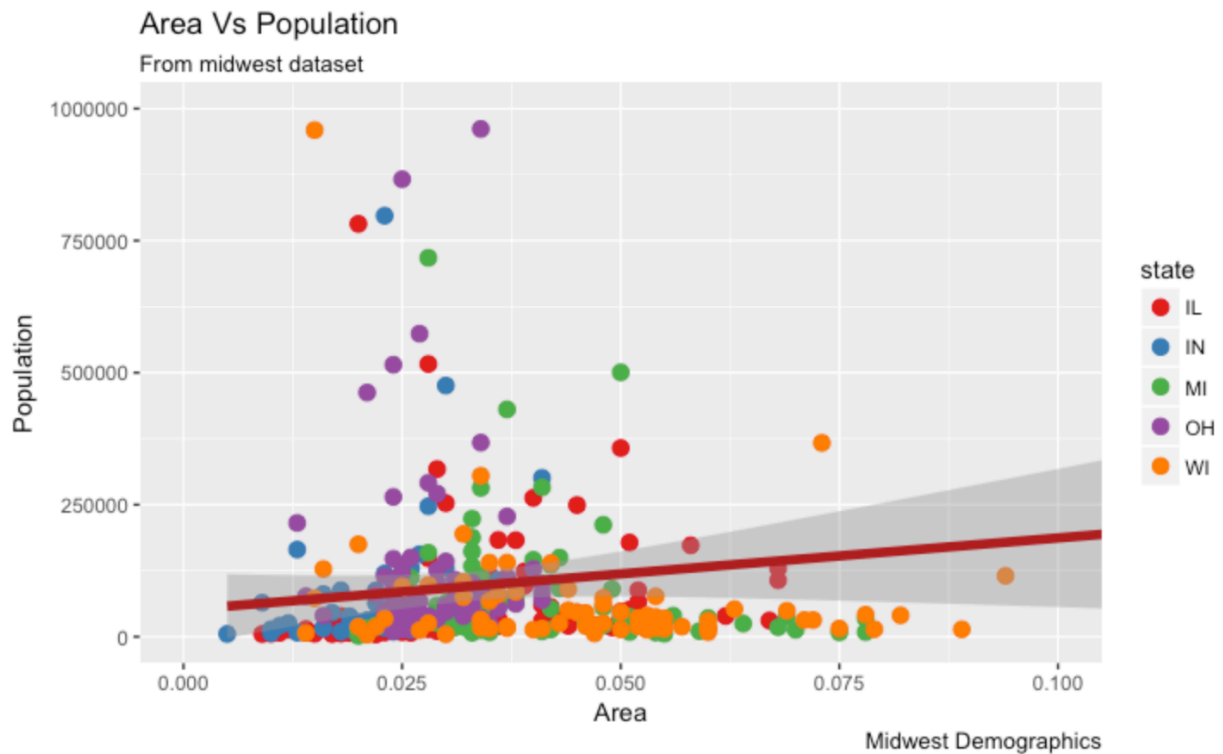
现在每个点都根据它所属的状态进行着色，因为 `aes (col = state)`。不仅可以使颜色，还可以使用尺寸，形状，笔划（边界厚度）和填充（填充颜色）来区分分组。

作为附加好处，图例自动添加。如果需要，可以通过在 `theme ()` 函数中将 `legend.position` 设置为 `None` 来删除它。

```
gg + theme(legend.position="None") # remove legend
```

另外，可以完全更改调色板。

```
gg + scale_colour_brewer(palette = "Set1") # change color palette
```



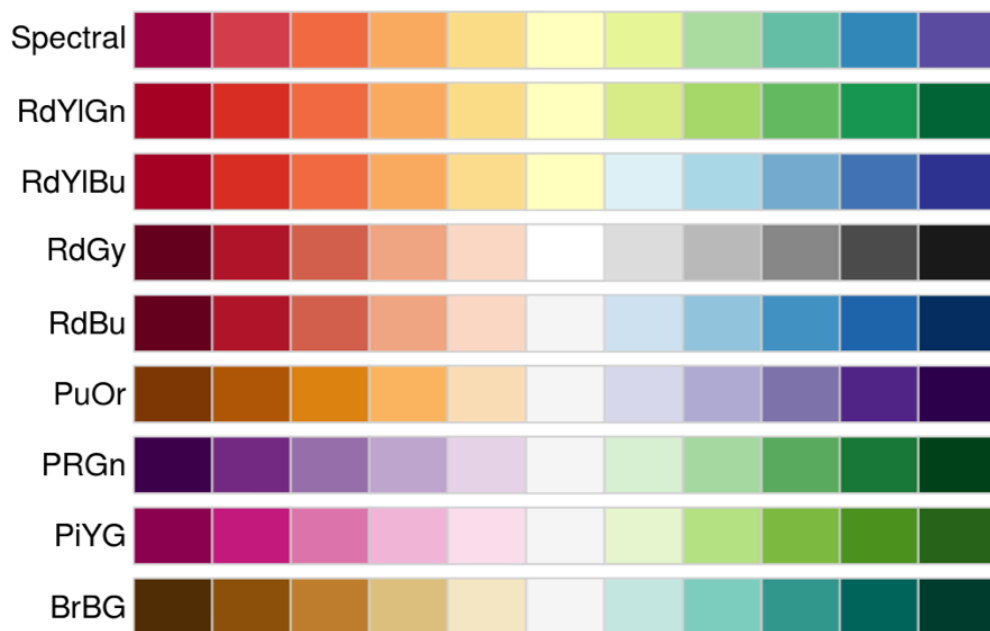
更多的这种调色板可以在 RColorBrewer 软件包中找到

```
library(RColorBrewer)
head(brewer.pal.info, 10) # show 10 palettes

#>      maxcolors category colorblind
#> BrBG      11    div    TRUE
#> PiYG      11    div    TRUE
#> PRGn      11    div    TRUE
#> PuOr      11    div    TRUE
#> RdBu      11    div    TRUE
#> RdGy      11    div   FALSE
#> RdYIBu    11    div    TRUE
#> RdYIGn    11    div   FALSE
```


#> Spectral	11	div	FALSE
#> Accent	8	qual	FALSE





6.如何更改 X 轴文本和位置

如何更改 X 和 Y 轴文本及其位置？

好的，现在来看看如何更改 X 轴和 Y 轴文本及其位置。这涉及两个方面：休息和标签。

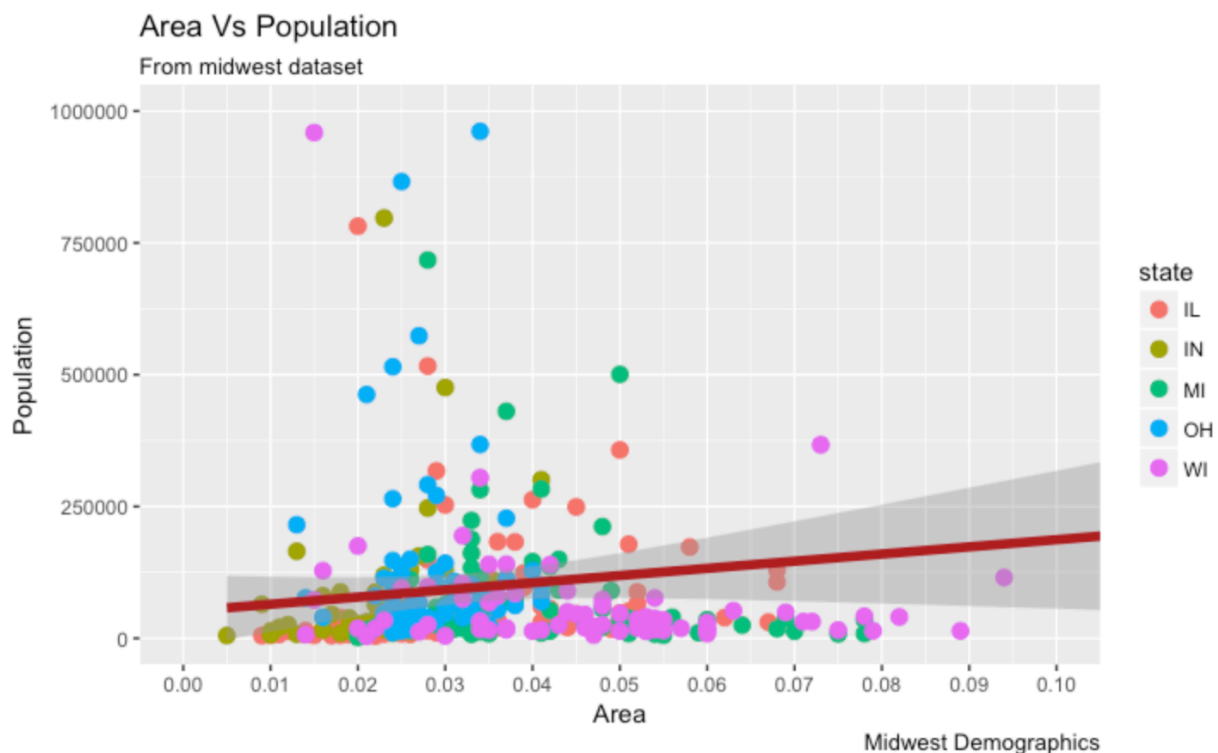
第 1 步：设置断点

中断应该与 X 轴变量具有相同的比例。请注意，我正在使用 `scale_x_continuous`，因为 X 轴变量是一个连续变量。如果它是一个日期变量，可以使用 `scale_x_date`。像 `scale_x_continuous()` 一样，`scale_y_continuous()` 可用于 Y 轴。

```
library(ggplot2)

# Base plot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state), size=3) + # Set color to vary based on state categories.
  geom_smooth(method="lm", col="firebrick", size=2) +
  coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
  labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Area", caption="Midwest Demographics")

# Change breaks
gg + scale_x_continuous(breaks=seq(0, 0.1, 0.01))
```



步骤 2：更改标签您可以选择更改轴标记处的标签。标签采用与休息时间长度相同的矢量。

让我通过将标签设置为从 a 到 k 的字母来演示（尽管在此上下文中它没有意义）。

```
library(ggplots)
```

```
# Base Plot
```

```
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
```

```
  geom_point(aes(col=state), size=3) + # Set color to vary based on state categories.
```

```
  geom_smooth(method="lm", col="firebrick", size=2) +
```

```
  coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
```

```
  labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Area", caption="Midwest Demographics")
```

```
# Change breaks + label
```

```
gg + scale_x_continuous(breaks=seq(0, 0.1, 0.01), labels = letters[1:11])
```

