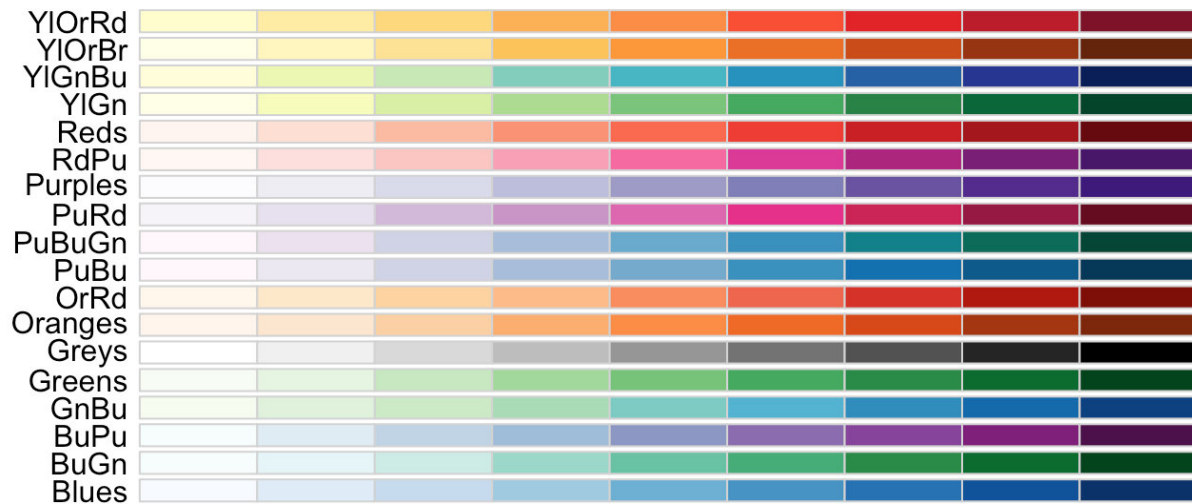


RColorBrewer

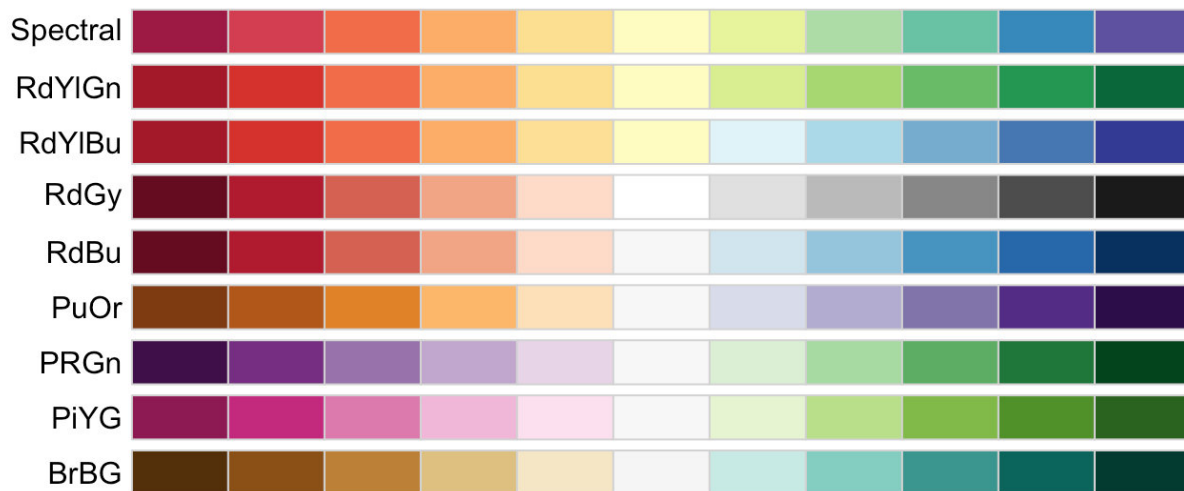
- RColorBrewer提供了3套配色方法

1. 连续型, sequential, 颜色渐变
2. 极端型, diverging, 生成深色强调两端, 浅色表示中部颜色, 可用来标注数据中的离群点
3. 离散型 qualitative, 生成彼此差异明显的颜色, 通常用来标记分类数据

```
display.brewer.all(type="seq")
```



```
display.brewer.all(type="div")
```



```
display.brewer.all(type="qual")
```



- 用法简单

`barplot(rep(1,6),col=brewer.pal(9,"Blues"))`# 数字9，表示使用色条中颜色的个数，引号内表示色条对应的名称，bar过多则依次循环使用配色

`barplot(rep(1,11),col=brewer.pal(11,"RdGy"))`

`barplot(rep(1,6),col=brewer.pal(11,"RdGy")[2:7])`# 第二到第七个颜色。

注意：仅有以上种类颜色，不适合连续变量使用

colorRamp/colorRampPalette函数(grDevices package)

colorRamp/colorRampPalette函数支持自定义的创建一系列颜色梯度

Color interpolation

Description

These functions return functions that interpolate a set of given colors to create new color palettes (like `topo.colors`) and color ramps, functions that map the interval $[0, 1]$ to colors (like `grey`).

Usage

```
colorRamp(colors, bias = 1, space = c("rgb", "Lab"),
           interpolate = c("linear", "spline"), alpha = FALSE)
colorRampPalette(colors, ...)
```

Arguments

<code>colors</code>	colors to interpolate; must be a valid argument to <code>col2rgb()</code> .
<code>bias</code>	a positive number. Higher values give more widely spaced colors at the high end.
<code>space</code>	a character string; interpolation in RGB or CIE Lab color spaces.
<code>interpolate</code>	use spline or linear interpolation.
<code>alpha</code>	logical: should alpha channel (opacity) values be returned? It is an error to give a true value if <code>space</code> is specified.
<code>...</code>	arguments to pass to <code>colorRamp</code> .

```
col <- colorRampPalette(c("red", "blue"))(10) 10色渐变
```

To be continued~

Here's a solution using base R graphics:

```
#Some sample data
x <- runif(100)
dat <- data.frame(x = x,y = x^2 + 1)

#Create a function to generate a continuous color palette
rbPal <- colorRampPalette(c('red','blue'))

#This adds a column of color values
# based on the y values
dat$Col <- rbPal(10)[as.numeric(cut(dat$y,breaks = 10))]

plot(dat$x,dat$y,pch = 20,col = dat$Col)
```

[搭配ggplot2使用][<https://www.cnblogs.com/shaocf/p/9600340.html>]

RColorBrewer为我们提供了一种通过使用构造函数colorRampPalette插入现有调色板来生成更大调色板的方法。它生成实际工作的函数：它们通过插入现有的调色板来构建具有任意数量颜色的调色板。要将调色板Set1插入到22种颜色（颜色数量存储在colourCount变量中，以供后续示例）

首先计算所需颜色数目

```
colourCount <- length(unique(mtcars$hp))
```

将调色板Set1插入到以上数目颜色中，供后续使用，构造函数getPalette

```
getPalette <- colorRampPalette(brewer.pal(12,"Set3")) #12取决于"Set3"
```

绘制

```
ggplot(mtcars) + geom_bar(aes(factor(hp)),fill=getPalette(colourCount)) +
scale_fill_manual(values=getPalette(colourCount))+
theme(legend.position="right")
```

