

庄闪闪的可视化手册

庄亮亮

2022 年 2 月 5 日

目录

简介	5
1 基础包	7
1.1 绘制基本图形	7
1.2 修改图形参数	24
1.3 绘制组合图形	38
1.4 保存图形	41
2 使用 ggplot2 包绘图	45
2.1 简介	45
2.2 qplot	45
2.3 ggplot2 包图形语法	51
2.4 几何对象	60
2.5 统计变换	62
2.6 刻度 scale	64
2.7 坐标系	71
2.8 分面	74
2.9 标题、标注、指南、拼接	78
2.10 保存图片	84
2.11 例子	84
3 其他相关拓展包	89
3.1 ggplot 官网 108 种拓展包	89
3.2 我整理的 11 个扩展包	89
4 认识交互式绘图工具	91
4.1 leaflet 包	92
4.2 dygraphs 包	92
4.3 plotly 包	92

4.4 DT 包	93
4.5 networkD3 包	93
4.6 利用 Shiny 包实现可交互的 Web 应用（待补充-更新可见公众号）	93
5 附录	95
安装 R 和 Rstudio	95
可能的问题	97
如何获取帮助	101
R 语言社区	102
6 中文图书 Bookdown 模板的基本用法	103
6.1 安装设置	103
6.2 编写自己的内容	103
6.3 转换	105
7 参考书籍	109

简介

R 软件的 bookdown 扩展包是 R Markdown 的增强版，支持自动目录、文献索引、公式编号与引用、定理编号与引用、图表自动编号与引用等功能，可以作为 LaTeX 的一种替代解决方案，在制作用 R 进行数据分析建模的技术报告时，可以将报告文字、R 程序、文字性结果、表格、图形都自动地融合在最后形成的网页或者 PDF 文件中。

Bookdown 使用的设置比较复杂，对初学者不够友好。这里制作了一些模板，用户只要解压缩打包的文件，对某个模板进行修改填充就可以变成自己的中文图书或者论文。Bookdown 的详细用法参见<https://bookdown.org/yihui/bookdown/>，在李东风的《统计软件教程》也有部分介绍。

一些常用功能的示例在 `0101-usage.Rmd` 文件中，用户可以在编辑器中打开此文件参考其中的做法。

Bookdown 如果输出为网页，其中的数学公式需要 MathJax 程序库的支持，用如下数学公式测试浏览器中数学公式显示是否正常：

$$\text{定积分} = \int_a^b f(x) dx$$

如果显示不正常，可以在公式上右键单击，选择“Math Settings–Math Renderer”，依次使用改成“Common HTML”，“SVG”等是否可以变成正常显示。PDF 版本不存在这样的问题。

Chapter 1

基础包

1.1 绘制基本图形

1.1.1 绘制分布关系

数据的数字特征刻画了数据的主要特征，而对数据总体情况做全面描述时，研究人员需要研究数据的分布情况。

主要方法：绘制相应的图形，如直方图、条形图、饼图、箱线图等。

1.1.1.1 直方图

概念介绍：直方图（Histogram）由一系列高度不等的纵向条纹或者线段表示数据分布的情况。

注意：一般用横轴表示数据所属类别，纵轴表示数量或者占比。

适用：连续数据。

例子：我们使用模拟数据进行讲解，通过正态分布产生 30 个数据。

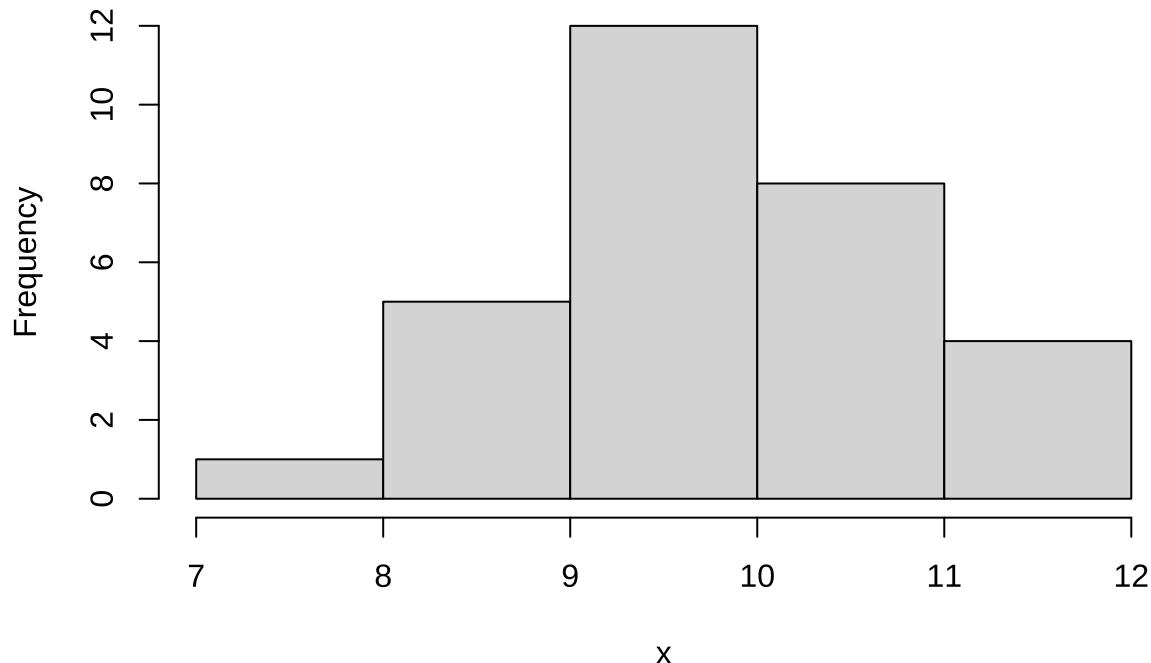
```
# 数据模拟产生
x <- rnorm(30, mean = 10, sd = 1)
print(round(x, 2))

## [1]  8.67 10.65  8.71  9.94  9.09 10.96 10.09  9.88 10.28 10.41 11.16  8.55
## [13] 10.62 11.19  9.82  9.85  9.89 10.81  9.73  9.60  8.58  9.83  9.61 10.55
## [25]  8.99 11.55  9.68 11.24  9.85  7.81
```

hist() 中的 breaks() 可以分段区间，取值可以是一个向量（各区间端点）或者一个数字（拆分为多少段），或者一个字符串（计算划分区间的算法名称），或者一个函数（划分区间个数的方法）。这里给出例子

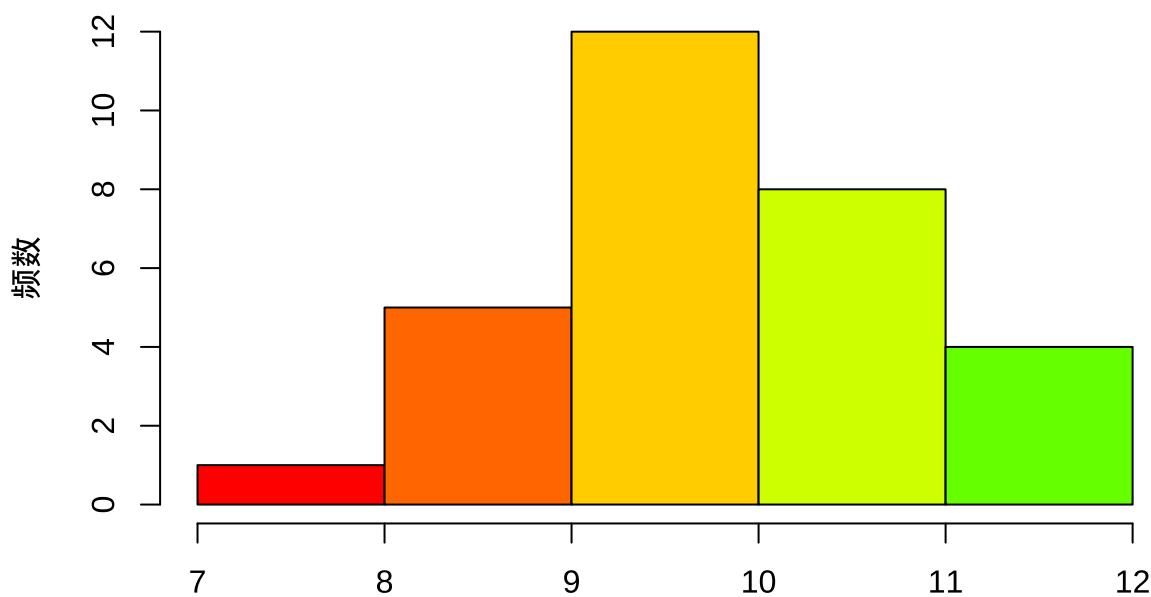
```
hist(x, breaks = 3)
```

Histogram of x



```
hist(x, col = rainbow(15), breaks = 3, main = " 正态随机数 ", xlab = "", ylab = " 频数 ")
```

正态随机数

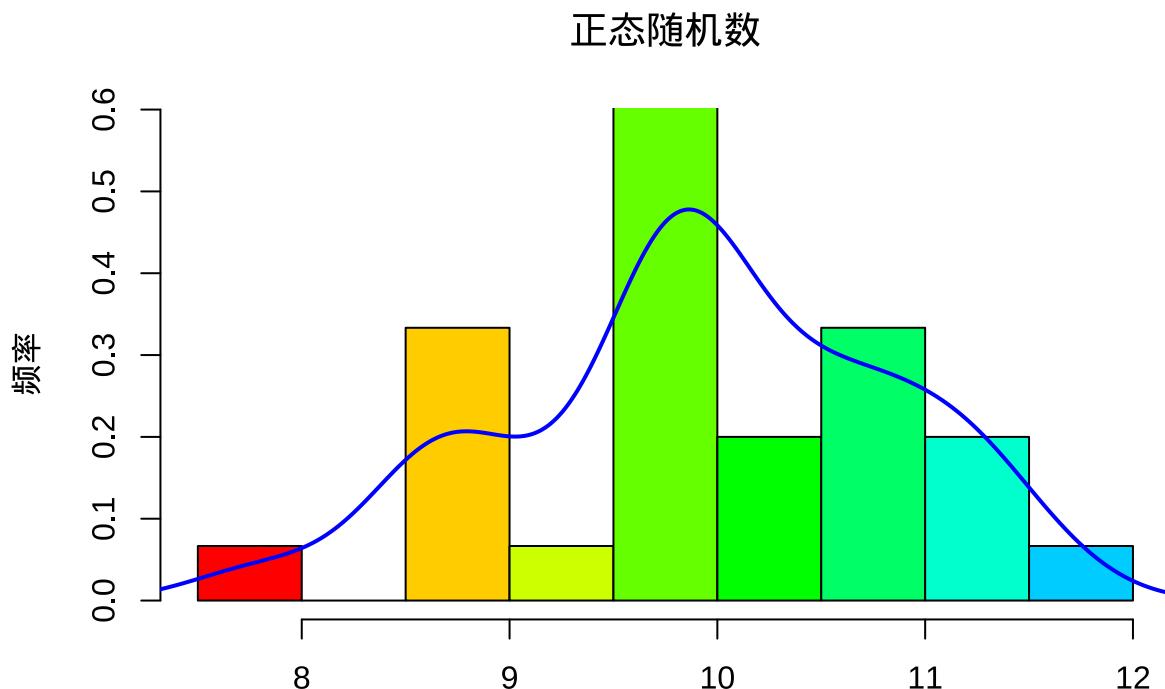


`breaks = 3` 表示 x 轴分为 3 个节点。其他设置可参考帮助文档, 即?hist。这里加入其他参数 `col`, `main`, `xlab`,

`ylab`, 分别表示颜色, 主题名称, x 轴名称, y 轴名称设置。细节将会在下面一章进行详细解释。

函数 `density()` 估计核密度。`freq=FALSE` 绘制频率图。下面的程序作直方图, 并使用 `lines()` 函数添加核密度曲线:

```
tmp.dens <- density(x)
hist(x, freq = FALSE, ylim = c(0, max(tmp.dens$y) + 0.1), col = rainbow(15), main = " 正态随机数",
      xlab = "", ylab = " 频率")
lines(tmp.dens, lwd = 2, col = "blue")
```



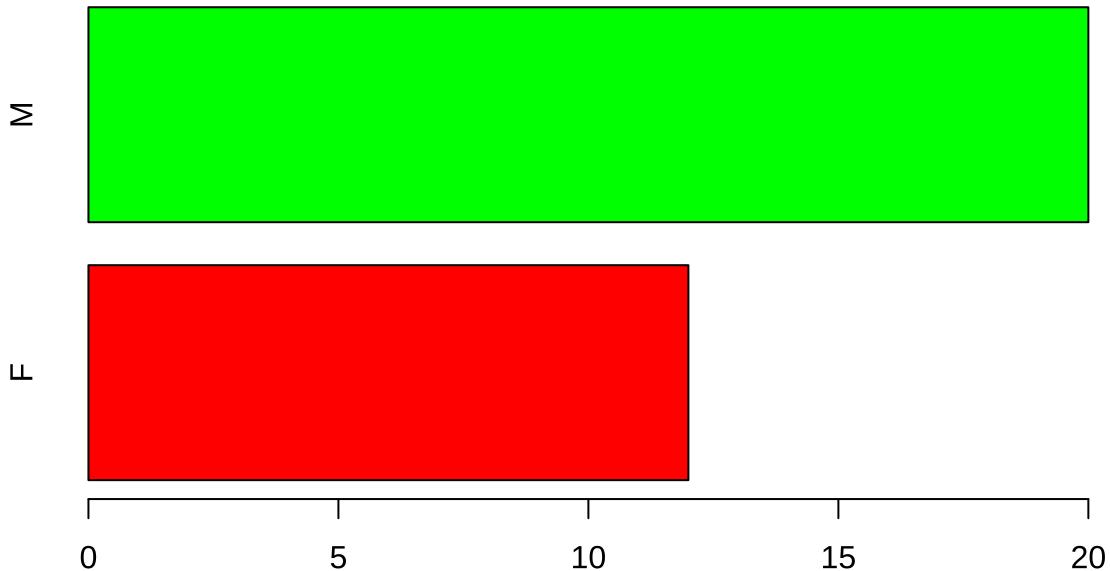
1.1.1.2 条形图

概念介绍: 数量的多少画成长短不同的直条, 然后把这些直条按一定的顺序排列起来。

注意: 条形图的 x 轴是数据类别 (离散型), y 轴是相应类别的频数。

```
# 复现课件中的条形图
gender = table(c(rep("F", 12), rep("M", 20)))
barplot(gender, col = c("red", "green"), main = " 性别分布", horiz = T)
```

性别分布



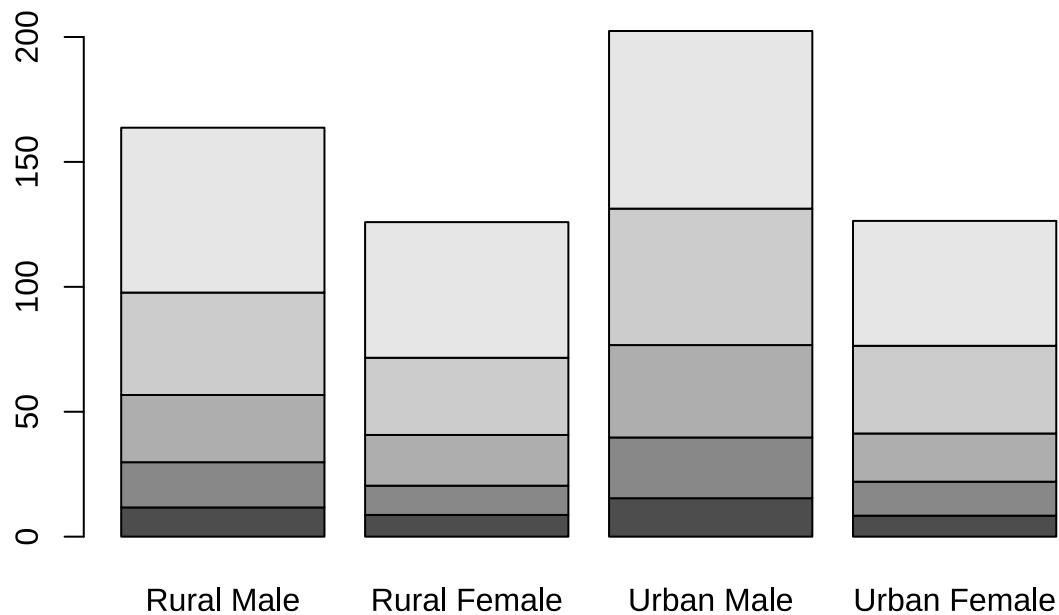
数据介绍: VADeaths 数据集记录的是 1940 年 Viginia(弗吉尼亚洲) 不同人群 (Rural Male、Rural Female、Urban Male、Urban Female) 中每一千人的死亡情况。

例子: 数据前 6 行展示如下:

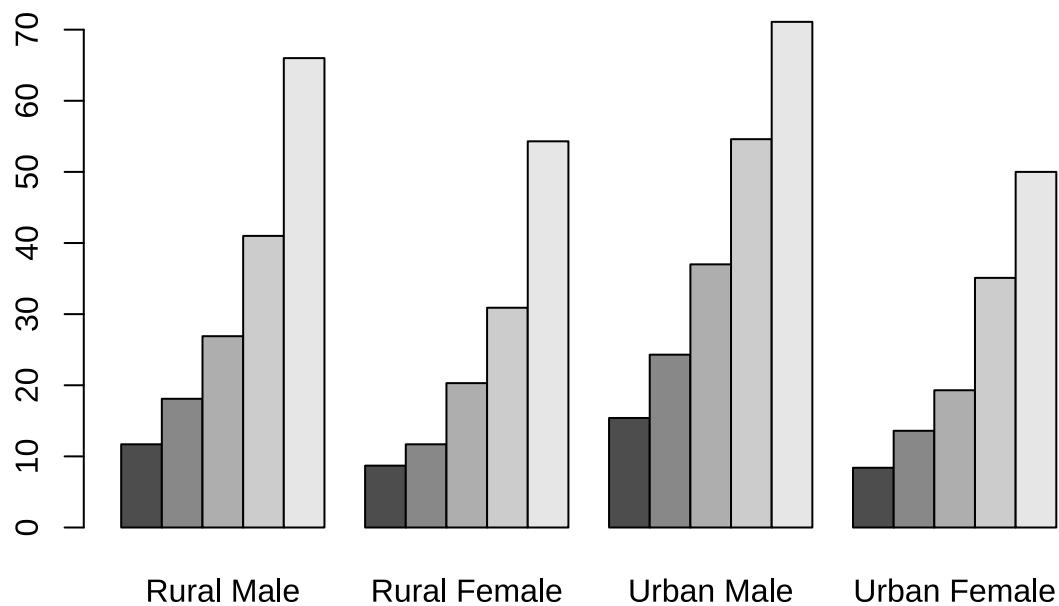
	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

这里绘制该数据的条形图。`beside` 默认值为 FALSE, 每一列都将给出堆砌的“子条”高度, 若 `beside=TRUE`, 则每一列都表示一个分组并列

```
barplot(VADeaths)
```



```
barplot(VADeaths, beside = TRUE)
```



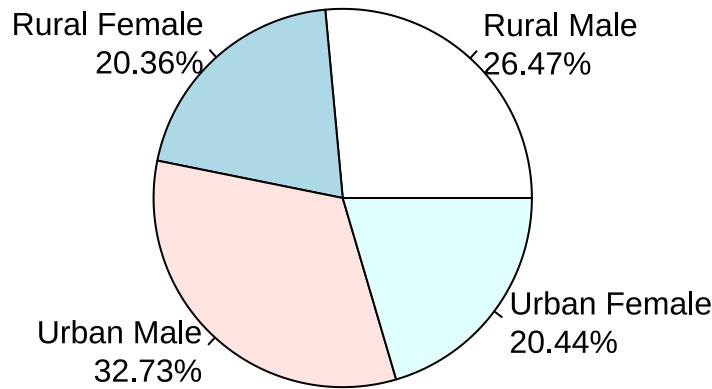
结论：随着年龄的增长，Virginia 人群的死亡率逐渐增加，并且在 4 类人群中，Urban Male 的死亡率比同年龄段的其他群体的死亡率高。同时，在同一环境下，相同年龄段的男性的死亡率要比女性高。

1.1.1.3 饼图

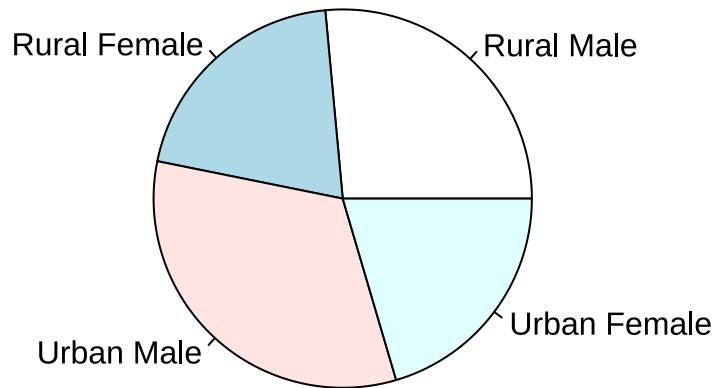
概念介绍：将各项的大小与各项总和的比例。反映部分与部分、部分与整体之间的比例关系。

例子：

```
percent <- colSums(VADeaths) * 100/sum(VADeaths)
pie(percent, labels = paste0(colnames(VADeaths), "\n", round(percent, 2), "%"))
```



```
pie(percent, radius = 0.8) #init.angle
```



```
# ?pie
```

结论：Virginia 人群中死亡最高的是 Urban Male，而且男性的死亡率比女性死亡率要高。

1.1.1.4 箱线图

概念介绍：绘制须使用常用的统计量（最小值、下四分位数、中位数、上四分位数和最大值），并提供有关数据位置和分散情况的关键信息，尤其在比较不同特征时，更可表现其分散程度差异。

数据介绍：iris 数据集（鸢尾花数据集），是常用的分类实验数据集合，由 Fisher 在 1936 年收集整理。数据集包含了 150 个子数据集，分为 3 类（分别为 setosa、versicolor、virginica），每类 50 个数据，每个数据包含 4 个属性，即花萼长度 Sepal.Length、花萼宽度 Sepal.Width、花瓣长度 Petal.Length、花瓣宽度 Petal.Width。

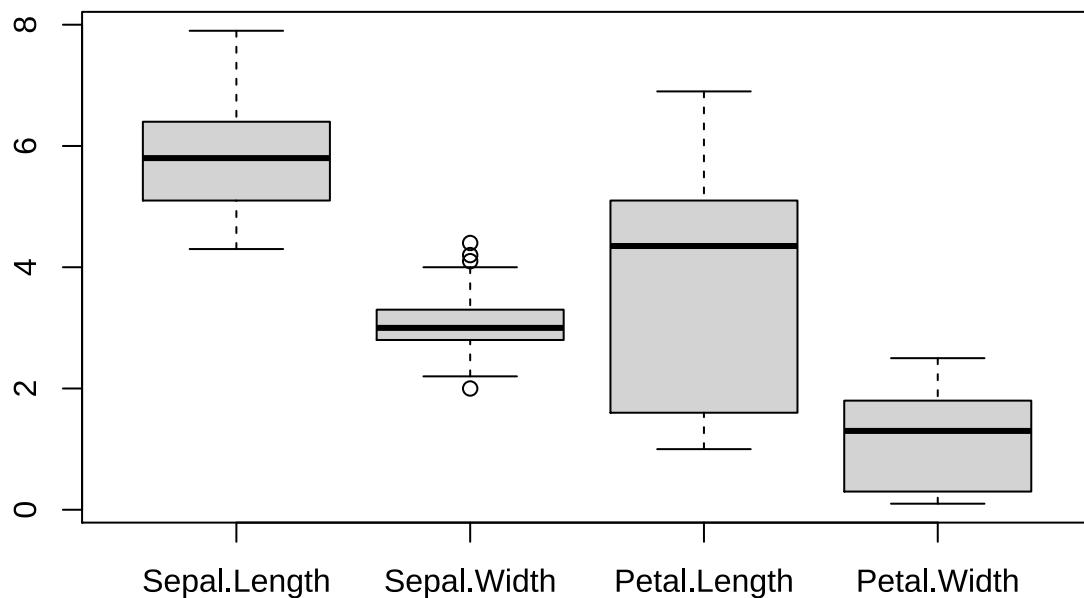
前 6 行数据如下：

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

例子：使用箱线图进行分析，使用两种方法：单独分析四个变量内部的数据分布情况；组间比较（Sepal.Length ~ Species）注意这里的 x 应该是因子型。这里没有对其他参数进行添加，大家根据自己需求添加即可。

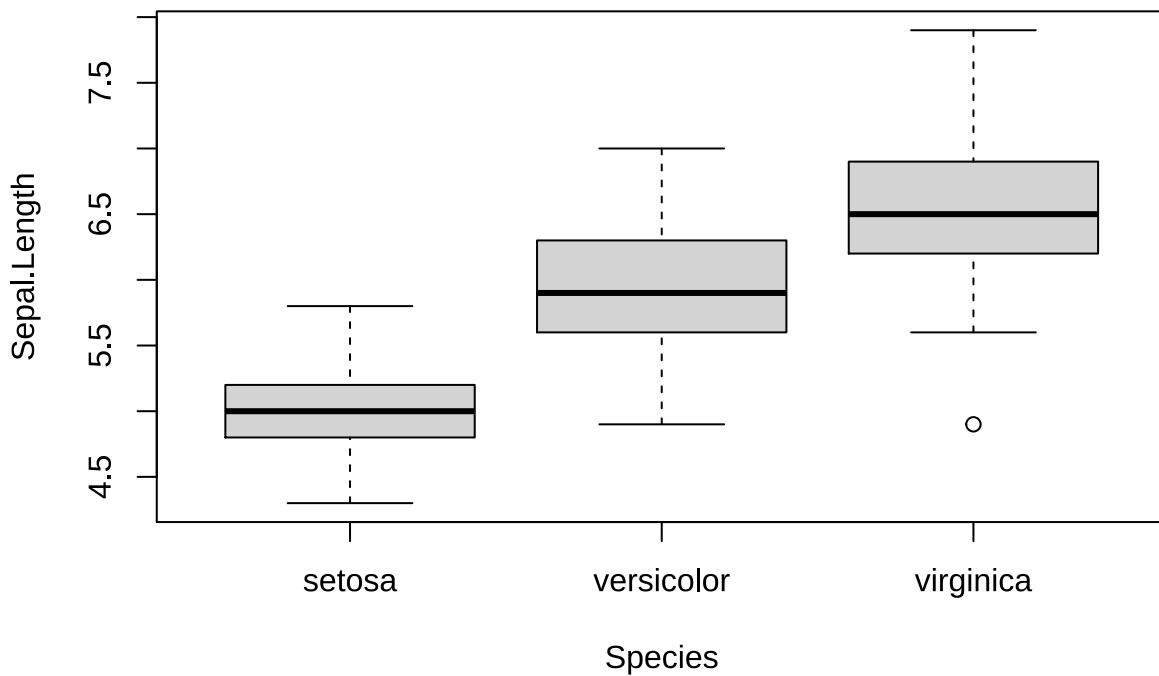
```
attach(iris)
boxplot(iris[1:4], main = " 单独的箱线图")
```

单独的箱线图



```
boxplot(Sepal.Length ~ Species, data = iris, main = " 组间比较的箱线图")
```

组间比较的箱线图



结论：第一个图：Sepal.Width 列含有四个异常值。

第二个图：Sepal.Length 列中，类别属于 virginica 的数据含有一个异常值。同时，从第一个图可以看到，Petal.Length 列前半部分相对分散，而后半部分相对密集。

1.1.2 绘制数据间关系

概念介绍：在分析数据间关系时，常用散点图和多变量相关矩阵图查看数据间的相关关系。

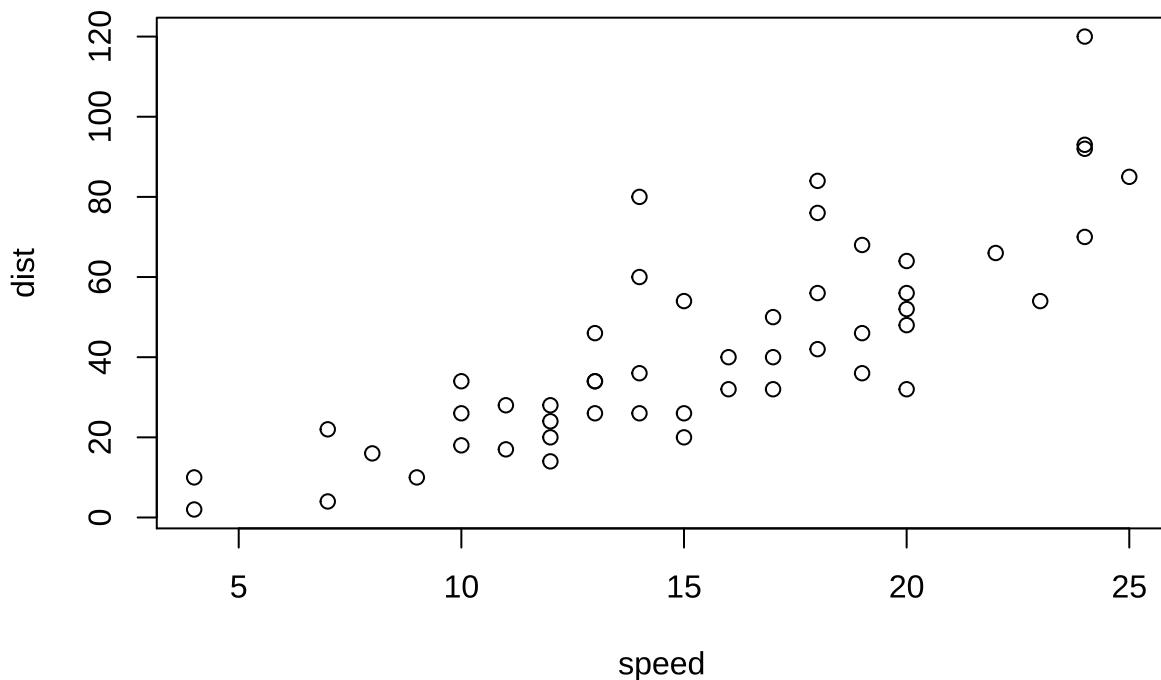
1.1.2.1 散点图

特点：

- (1) 特征之间是否存在关联趋势，关联趋势是线性的还是非线性的。
- (2) 一目了然的看出离群值。从而可以进一步分析这些离群值是否可能在建模分析中产生很大的影响。

例子：使用 cars 数据进行分析速度 (speed) 和刹车距离 (dist) 之间的关系

```
plot(cars[, 1], cars[, 2], xlab = "speed", ylab = "dist")
```



```
# plot(cars) # 效果同上
```

结论：随着汽车行驶速度的增加，刹车距离也在不断增加。

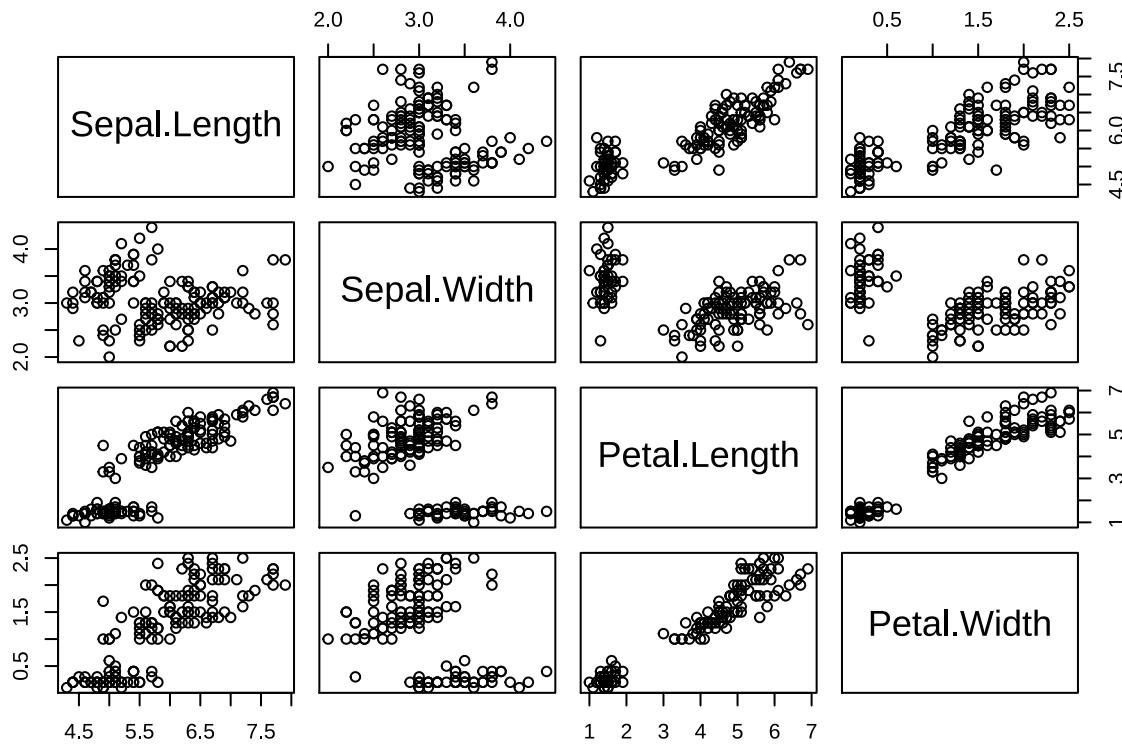
1.1.2.2 散点矩阵图

概念介绍：散点矩阵图将多个散点图组合起来，以便可以同时浏览多个二元变量关系，一定程度上克服了在平面上展示高维数据分布情况的困难。可以使用 `plot()` 或者 `pairs()` 进行绘制。

适用：高维数据

- `plot`

```
plot(iris[, 1:4])
```

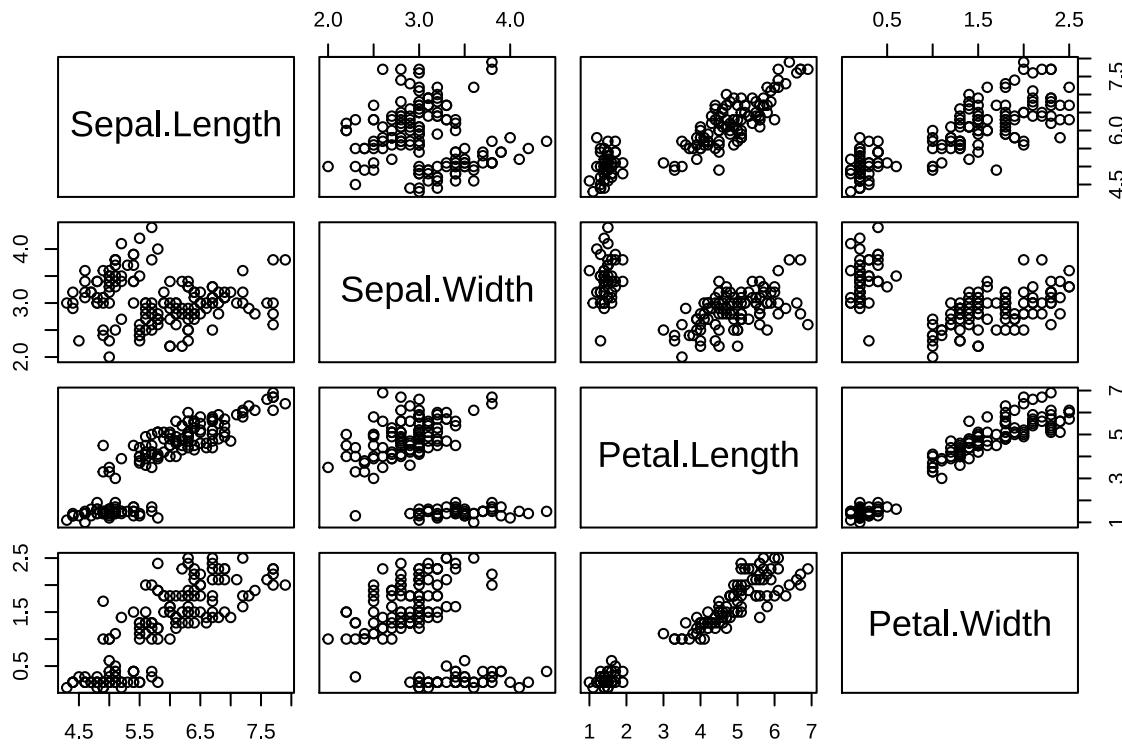


结论：花瓣长度（Petal.length）与花瓣宽度（Petal.Width）有明显的线性关系，其余属性之间的关系不是很明显。

- pairs

此外，R 中还提供了另一个绘制散点矩阵图的函数——pairs 函数，绘图对象有数据框和公式两种：

```
pairs(iris[, 1:4])
pairs(~Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = iris) # 效果同上
```



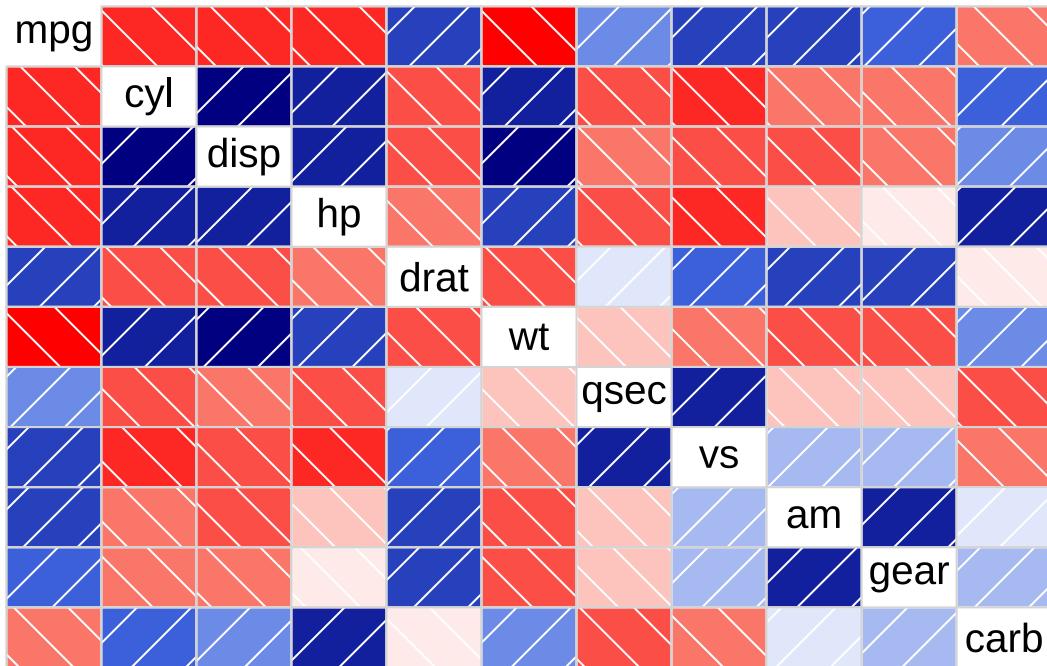
1.1.2.3 多变量相关矩阵图

概念介绍：多变量相关矩阵图是相关系数矩阵（correlation matrix）的可视化结果，显示了两两变量间的关系，对数据维度相对较大的数据有较好的展示效果。在 R 的 `corrgram` 包中的 `corrgram` 函数可绘制多变量相关矩阵图。

数据介绍：Mtcars 数据集是 1974 年 Motor Trend US 杂志公布的 32 辆车的 11 个数据，包括燃料消耗和 10 个关于汽车设计与性能的数据。

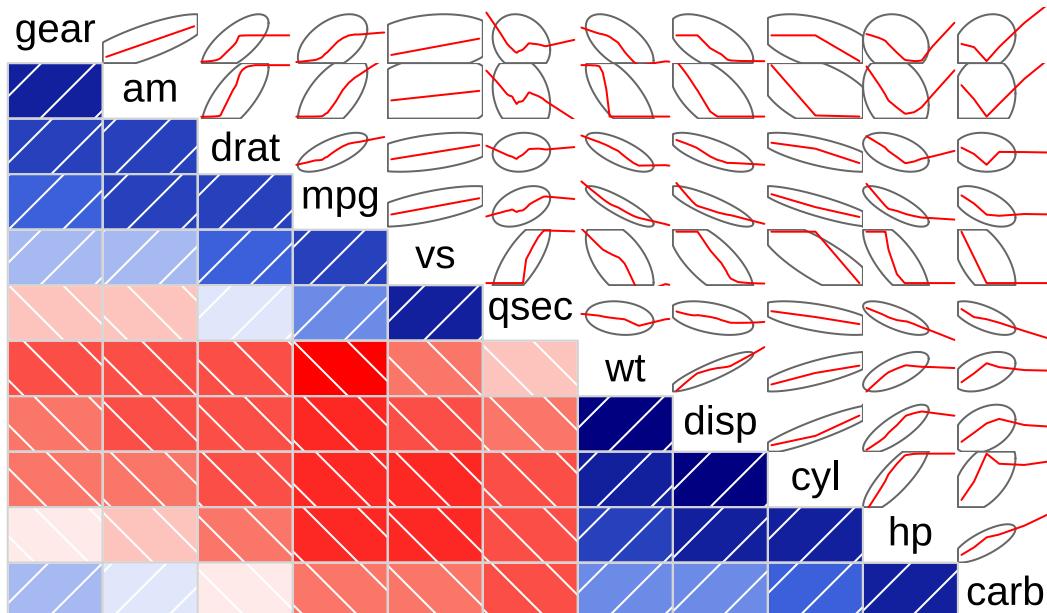
例子：下面、根据这个数据集，绘制适用中不同元素描述相关性大小的图。

```
library(corrgram)
corrgram(mtcars)
```



```
corrgram(mtcars, order = TRUE, upper.panel = panel.ellipse, main = "Correlogram of mtcars intercorrel")
```

Correlogram of mtcars intercorrelations

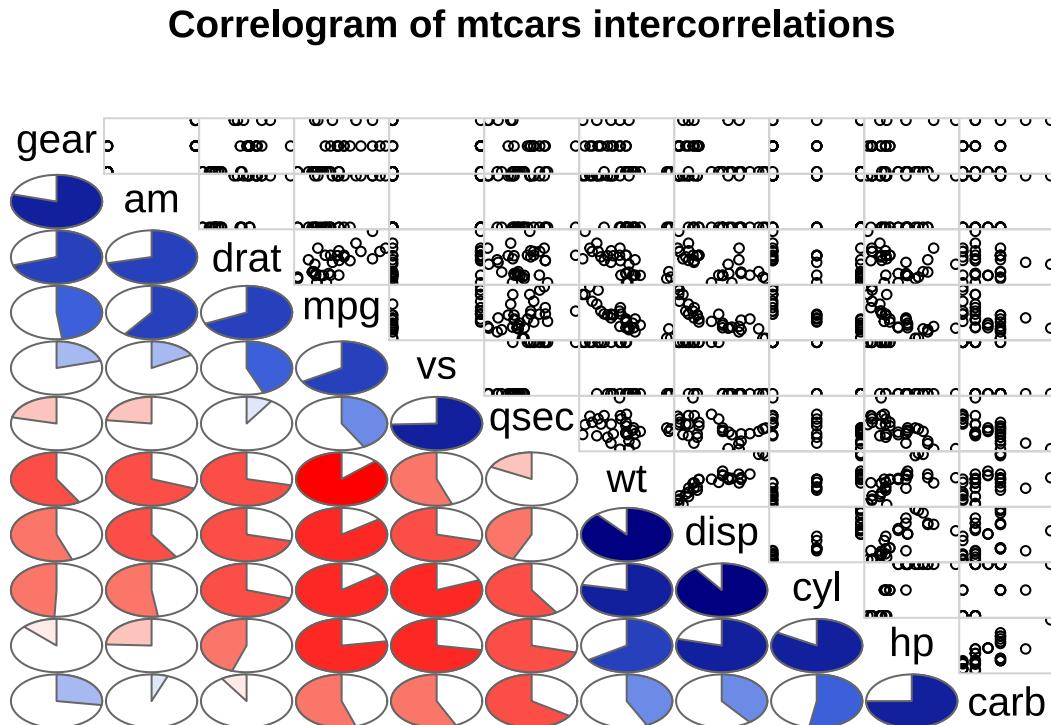


```
# 相关图，主对角线上方绘制置信椭圆和平滑拟合曲线，主对角线下方绘制阴影
```

结论: disp 与 cyl 呈正相关关系, 且相关程度较高。此外, mpg 与 wt 呈高度负相关, 且 am 与 carb 基本没有关系。

这里可以对 `upper.panel` 和 `lower.panel` 进行设置，展示不同图形。具体可以通过帮助获得相信参数设置信息 (`?corrgram`)

```
# 相关图，主对角线上方绘制散点图，主对角线下方绘制饼图
corrgram(mtcars, order = TRUE, upper.panel = panel.pts, lower.panel = panel.pie,
         main = "Correlogram of mtcars intercorrelations")
```



```
# 相关图，主对角线上方绘制置信区间，主对角线下方绘制相关系数
corrgram(mtcars, order = TRUE, upper.panel = panel.conf, lower.panel = panel.cor,
         main = "Correlogram of mtcars intercorrelations")
```

```
corrgram(mtcars, order = TRUE, upper.panel = panel.conf, lower.panel = panel.cor,
         main = "Correlogram of mtcars intercorrelations")
```

Correlogram of mtcars intercorrelations

gear	0.79 (0.62,0.89)	0.70 (0.46,0.84)	0.48 (0.16,0.71)	0.21 (-0.15,0.52)	-0.21 (-0.52,0.15)	-0.58 (-0.77,-0.29)	-0.56 (-0.76,-0.26)	-0.49 (-0.72,-0.17)	-0.13 (-0.45,0.23)	0.27 (-0.08,0.57)
0.79 am	0.71 (0.48,0.85)	0.60 (0.32,0.78)	0.17 (-0.19,0.49)	-0.23 (-0.54,0.13)	-0.69 (-0.84,-0.45)	-0.59 (-0.78,0.31)	-0.52 (-0.74,-0.21)	-0.24 (-0.55,0.12)	0.06 (-0.30,0.40)	
0.70 0.71 drat	0.68 (0.44,0.83)	0.44 (0.11,0.68)	0.09 (-0.27,0.43)	-0.71 (-0.85,-0.48)	-0.71 (-0.85,0.48)	-0.70 (-0.84,-0.46)	-0.45 (-0.69,-0.12)	-0.09 (-0.43,0.27)		
0.48 0.60 0.68 mpg	0.66 (0.41,0.82)	0.42 (0.08,0.67)	0.42 (-0.93,0.74)	-0.87 (-0.92,0.71)	-0.85 (-0.93,0.72)	-0.85 (-0.89,-0.59)	-0.78 (-0.75,0.25)			
0.21 0.17 0.44 0.66 vs	0.74 (0.53,0.87)	-0.55 (-0.76,-0.26)	-0.71 (-0.85,-0.48)	-0.81 (-0.90,-0.64)	-0.72 (-0.86,-0.50)	-0.57 (-0.77,-0.28)				
-0.21 -0.23 0.09 0.42 0.74 qsec	-0.17 (-0.49,0.19)	-0.43 (-0.69,0.10)	-0.59 (-0.78,-0.31)	-0.71 (-0.85,-0.40)	-0.71 (-0.82,-0.40)	-0.66 (-0.82,-0.40)				
-0.58 -0.69 -0.71 -0.87 -0.55 -0.17 wt	0.89 (0.78,0.94)	0.78 (0.60,0.89)	0.66 (0.40,0.82)	0.43 (0.09,0.58)						
-0.56 -0.59 -0.71 -0.85 -0.71 -0.43 disp	0.89 (0.81,0.95)	0.89 (0.61,0.89)	0.90 (0.05,0.55)	0.79 (0.68,0.92)	0.53 (0.22,0.74)					
-0.49 -0.52 -0.70 -0.85 -0.81 -0.59 cyl	0.78 (0.51,0.87)	0.90 (0.68,0.92)	0.83 (0.22,0.74)	0.75 (0.51,0.87)						
-0.13 -0.24 -0.45 -0.78 -0.72 -0.71 hp	0.66 (0.51,0.87)	0.79 (0.68,0.92)	0.83 (0.22,0.74)	0.75 (0.51,0.87)						
0.27 0.06 -0.09 -0.55 -0.57 -0.66 carb	0.43 (0.51,0.87)	0.39 (0.68,0.92)	0.53 (0.22,0.74)	0.75 (0.51,0.87)						

1.1.3 绘制其他图形

1.1.3.1 核密度图

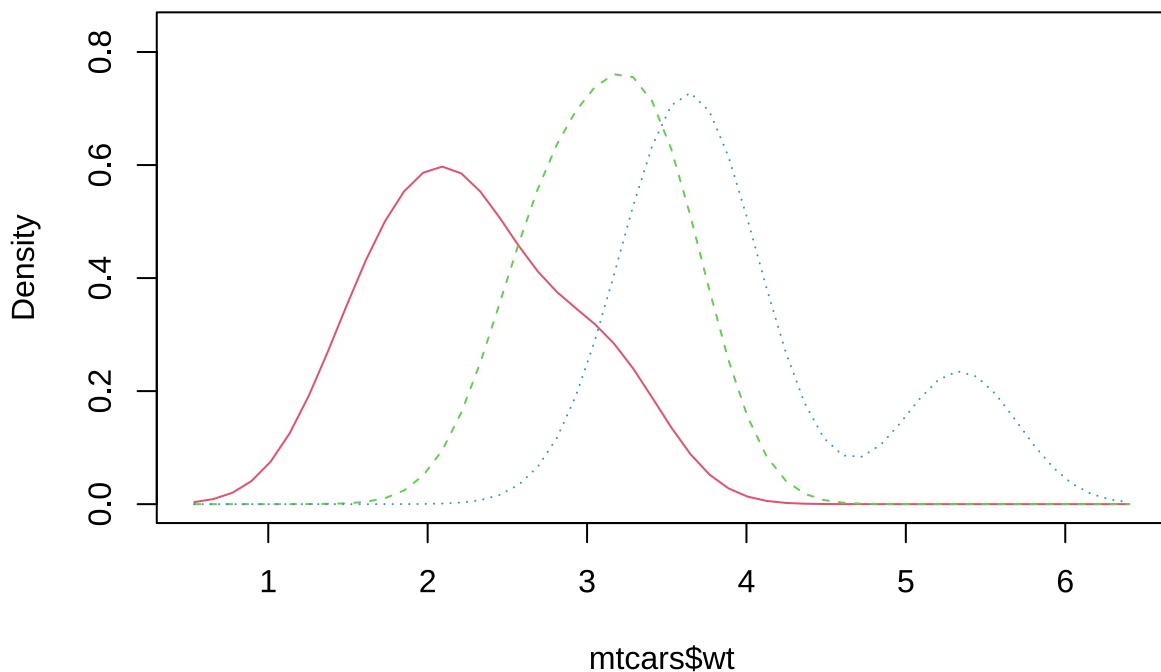
概念介绍：sm 包中 `sm.density.compare` 函数用于绘制核密度图，核密度图如果想用一条密度曲线而不是通过柱状来展示连续型变量的分布。

特点：相比直方图，密度图的一个优势是可以堆放，可用于比较组间差异。`sm.density.compare` 函数可以直接堆放多条密度曲线。使用格式如下。

```
sm.density.compare(x ,group,...)
```

其中 x 是数值向量，group 是分组向量，是因子型数据。

```
library(sm) # 加载 sm 包
sm.density.compare(mtcars$wt, factor(mtcars$cyl)) # 绘制核密度图
```



1.1.3.2 小提琴图

概念介绍：vioplot 包中的 vioplot 函数用于绘制小提琴图，小提琴图是核密度图与箱线图的结合，本质是利用密度值生成的多边形，但该多边形同时还沿着一条直线作了另一半对称的“镜像”，这样两个左右或上下对称的多边形拼起来就形成了小提琴图的主体部分，最后一个箱线图也会被添加在小提琴的中轴线上。

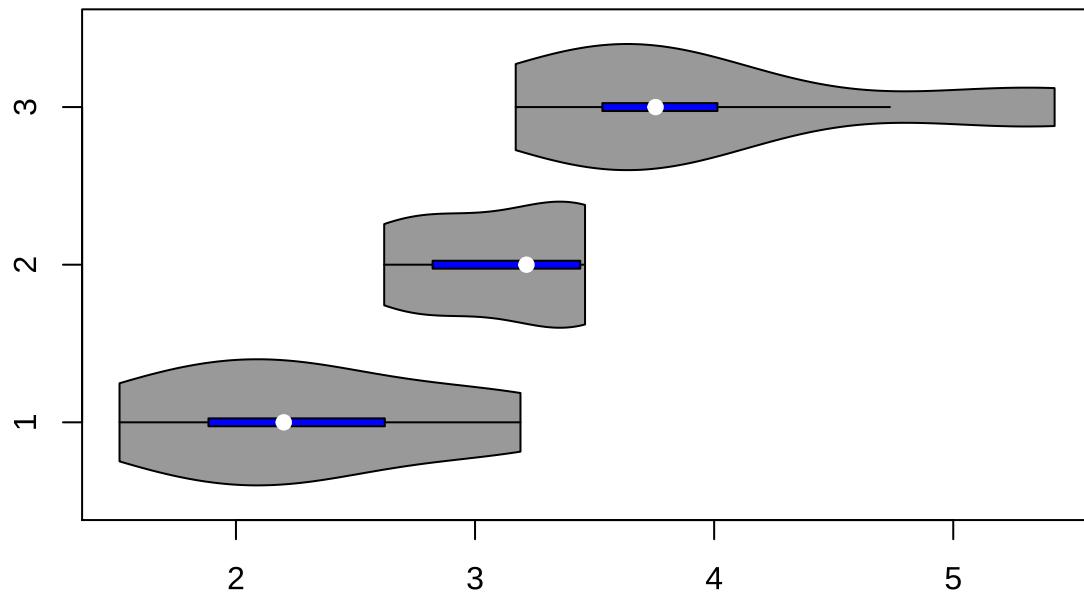
使用格式如下。

```
vioplot( x , ... , range=1.5, h, ylim, names, horizontal=FALSE , ...)
```

其中，x 为数据源，可以是向量；range 默认等于 1.5；col 是为每幅小提琴图指定颜色的向量。

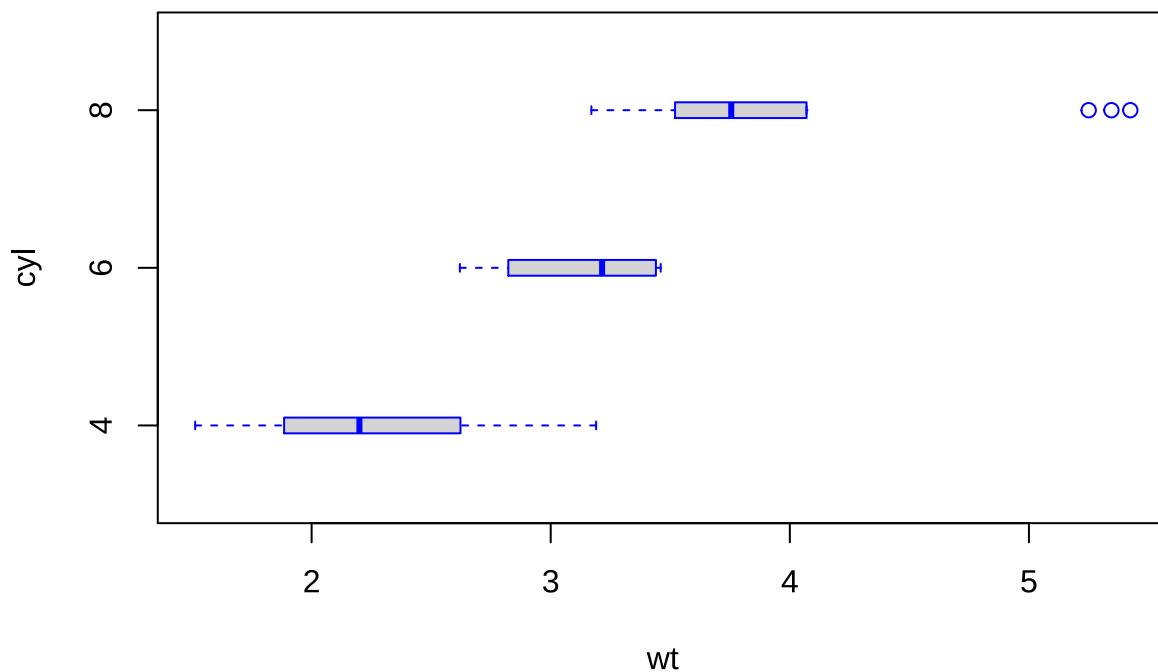
```
library(vioplot) # 加载 vioplot 包
attach(mtcars)
vioplot(wt[cyl == 4], wt[cyl == 6], wt[cyl == 8], border = "black", col = "gray60",
rectCol = "blue", horizontal = TRUE, main = " 小提琴图") # 绘制小提琴图
```

小提琴图



```
boxplot(wt ~ cyl, main = " 箱线图", horizontal = TRUE, pars = list(boxwex = 0.1),
        border = "blue") # 绘制箱线图
```

箱线图



1.1.3.3 QQ 图

概念介绍：查看经验分布和理论分布是否一致。将排序后的数据和理论分布的分位数进行比较后大致相等，说明了经验分布和理论分布相似。`qqplot()` 函数用于绘制 QQ 图，QQ 图检查数据是否服从某种分布。

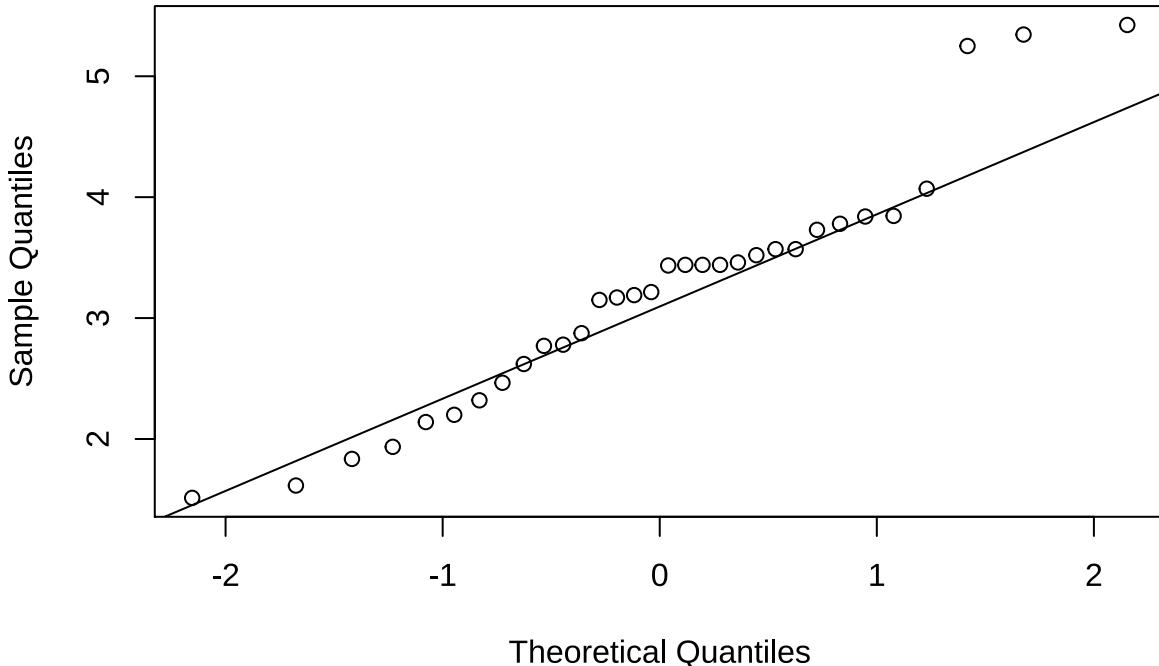
使用格式如下：

```
qqplot(x, y, ...); qqnorm(y, ...); qqline(y)
```

其中，`x` 与 `y` 均为数据源，可以是向量。

```
qqnorm(wt) # 正态分布 QQ 图  
qqline(wt) # QQ 线
```

Normal Q-Q Plot



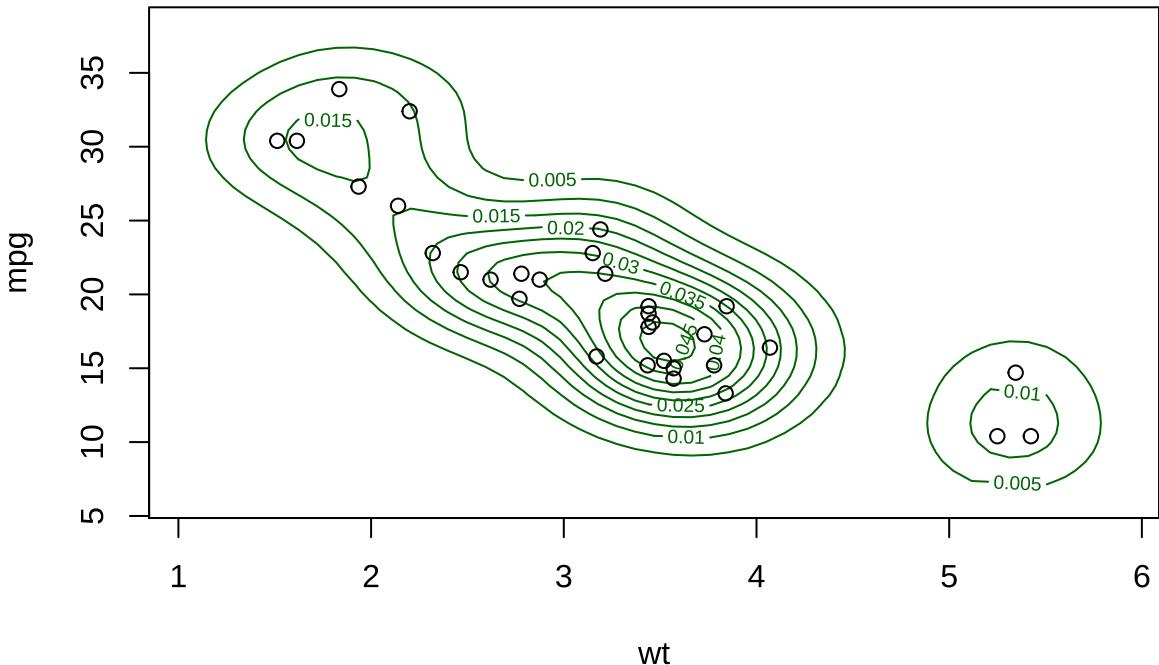
1.1.3.4 等高图

概念介绍：数据形式：两个数值向量 `x`、`y` 和一个相应的矩阵 `z`。`x`、`y` 交叉组合之后形成的是一个“网格”，`z` 是这个网格上的高度数值，将平面上对应的 `z` 值（高度）相等的点连接起来形成的线就是等高线。对 `x`、`y` 进行核密度估计，得到一个密度值矩阵，然后用 `x`、`y` 以及这个密度值矩阵作等高图。由于密度值反映的是某个位置上数据的密集程度，等高图就展示了一个聚类现象。

使用格式如下

```
contour(x=,y=,z,nlevels=,levels=,labels=,method=,...)
```

```
library(KernSmooth) # 计算二维核密度的包
mtcars1 = data.frame(wt, mpg)
est = bkde2D(mtcars1, apply(mtcars1, 2, dpik)) # 计算二维核密度
contour(est$x1, est$x2, est$fhat, nlevels = 15, col = "darkgreen", xlab = "wt", ylab = "mpg") # 画等
points(mtcars1) # 添加散点
```



1.2 修改图形参数

R 是一个功能强大的图形构建平台，可以通过逐条输入语句构建图形元素（颜色、点、线、文本以及图例等），逐渐完善图形特征，直至得到想要的效果。图形元素的显示可以用图形函数和绘图参数来改良，也可以用绘制图形元素的基础函数来控制。

1.2.1 修改颜色

R 语言通过设置绘图参数 `col` 来改变图像、坐标轴、文字、点、线等的颜色。关于颜色的函数大致可以分为下面三类：

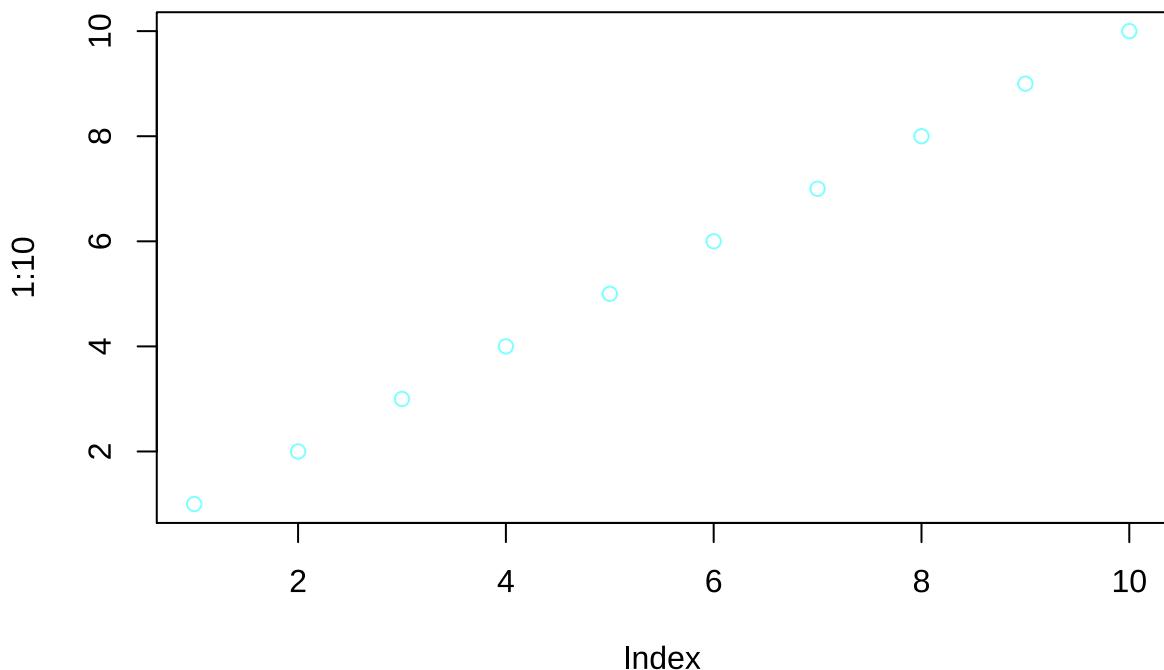
1.2.1.1 固定颜色选择函数

R 语言提供了自带的固定种类的颜色，主要涉及的是 `colors` 函数，该函数可以生成 657 中颜色名称，代表 657 种颜色，可以通过以下代码查看 R 自带颜色的前 20 中颜色的名称。

```
colors()[1:20]
```

```
## [1] "white"          "aliceblue"       "antiquewhite"    "antiquewhite1"
## [5] "antiquewhite2"  "antiquewhite3"  "antiquewhite4"  "aquamarine"
## [9] "aquamarine1"    "aquamarine2"    "aquamarine3"    "aquamarine4"
## [13] "azure"          "azure1"         "azure2"         "azure3"
## [17] "azure4"          "beige"          "bisque"         "bisque1"

# colors()
plot(1:10, col = cm.colors(1))
```



通过 palette 函数固定调色板，只要设定好了调色板，它的取值就不会再改变（直到下一次重新设定调色板）。

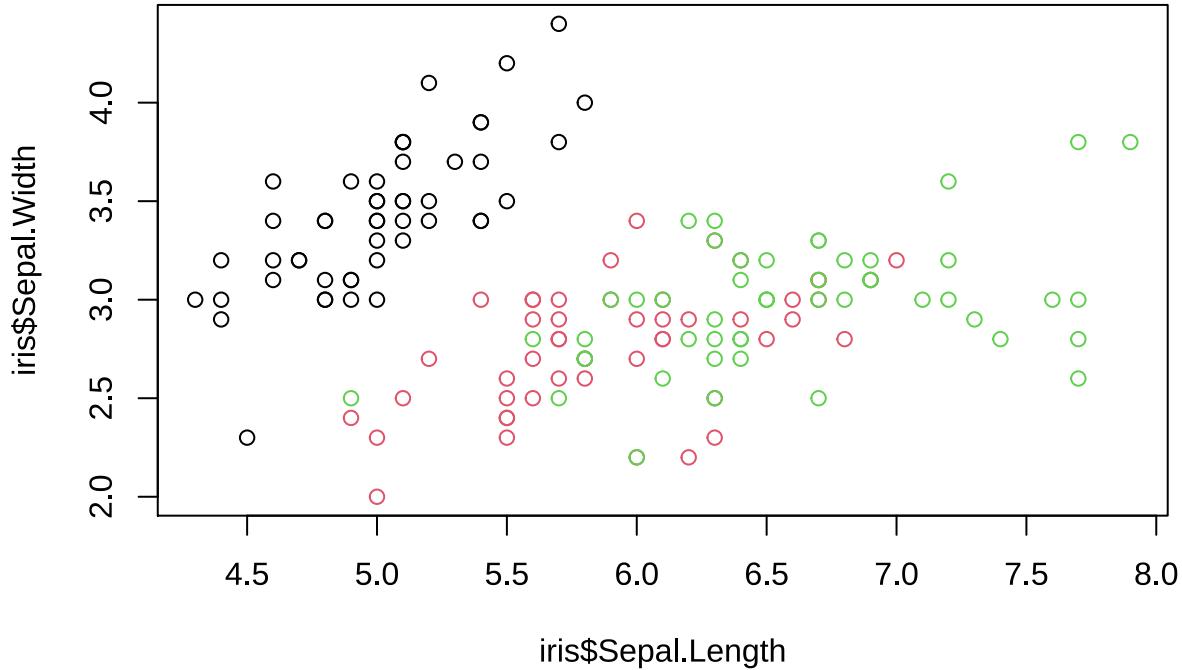
```
palette() # 返回当前的调色板设置，此时为默认值
## [1] "black"      "#DF536B"   "#61D04F"   "#2297E6"   "#28E2E5"   "#CDOBBC"   "#F5C710"
## [8] "gray62"

palette(colors()[1:10]) # 重新设置调色板为 colors 的前 10 种颜色
palette() # 返回当前的调色板设置，此时为 colors() 的前 10 种颜色
## [1] "white"        "aliceblue"     "antiquewhite"  "antiquewhite1"
## [5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"
## [9] "aquamarine"   "aquamarine2"

palette("default") # 恢复默认的调色板设置
```

例子：

```
plot(iris$Sepal.Length, iris$Sepal.Width, col = iris$Species)
```



```
# Species 为因子型数据, setosa versicolor virginica 分别对应数字 1, 2, 3,  
# 即等价于 col = rep(1:3, each = 50)
```

1.2.1.2 渐变色生成函数

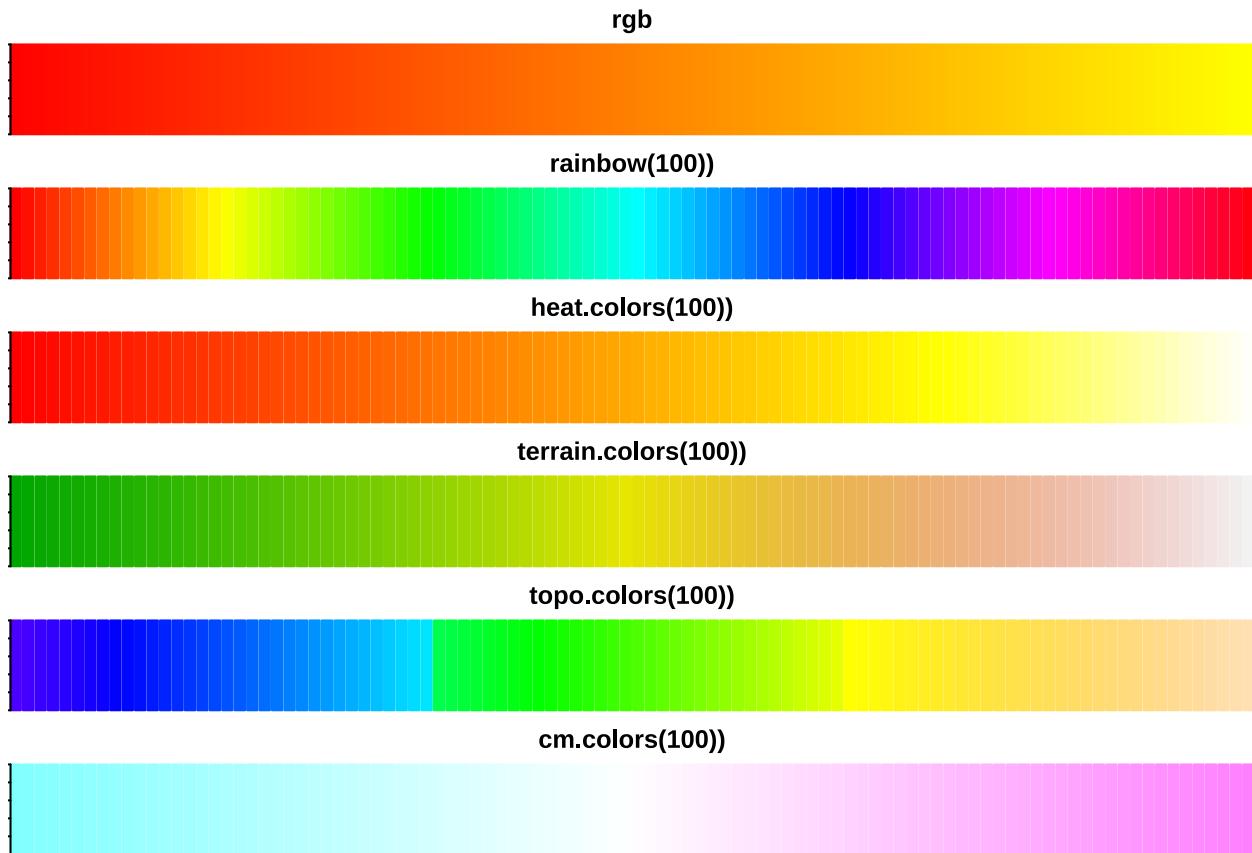
除了固定颜色选择函数外，R 还提供了一系列渐变颜色生成函数，这些函数用来控制颜色值逐步变化。

函数名称	生成原理	使用格式
rgb	RGB模型(红绿蓝混合)	rgb(red,green,blue,alpha, names=NULL,max=1)
rainbow	彩虹色(赤橙黄绿青蓝紫)	rainbow(n,s=1,v=1,start=0,end=max(1,n-1)/n,gamma=1)
heat.colors	高温、白热化(红黄白)	同 rainbow 函数
terrain.colors	地理地形(绿黄棕白)	同 rainbow 函数
topo.colors	蓝青黄棕	同 rainbow 函数
cm.colors	青白粉红	同 rainbow 函数
brewer.pal	RColorBrewer包提供的3套配色方案	col=brewer.pal(n,"颜色组*")) 颜色组*: 3类配色方案的颜色组名称

`rgb` 函数把 RGB 颜色转化为十六进制数值，使用格式前四个参数都取值于区间 [0, max]，`names` 参数用来指定生成颜色向量的名称。`red`, `green`, `blue` 参数的值越大就说明该颜色的成分越高。`alpha` 指的是颜色的透明度，取

0 表示完全透明，取最大值表示完全不透明（默认完全不透明）。

`rainbow` 函数、`heat.colors` 函数、`terrain.colors` 函数、`topo.colors` 函数、`cm.colors` 函数是主题配色函数，使用格式中 `n` 设定产生颜色的数目，`start` 和 `end` 设定彩虹颜色的一个子集，生成的颜色将从这个子集中选取。



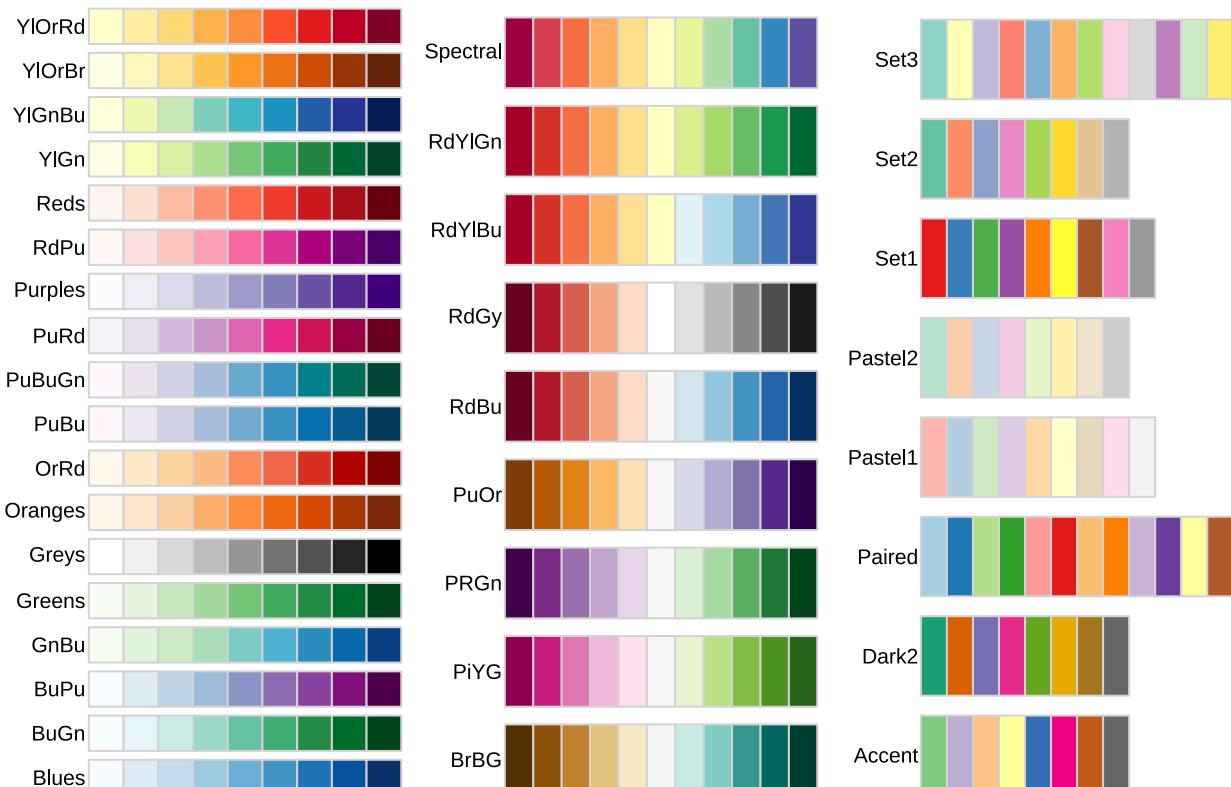
1.2.1.3 RColorBrewer 包

RColorBrewer 包提供了 3 套配色方案，分别为连续型，极端型以及离散型。

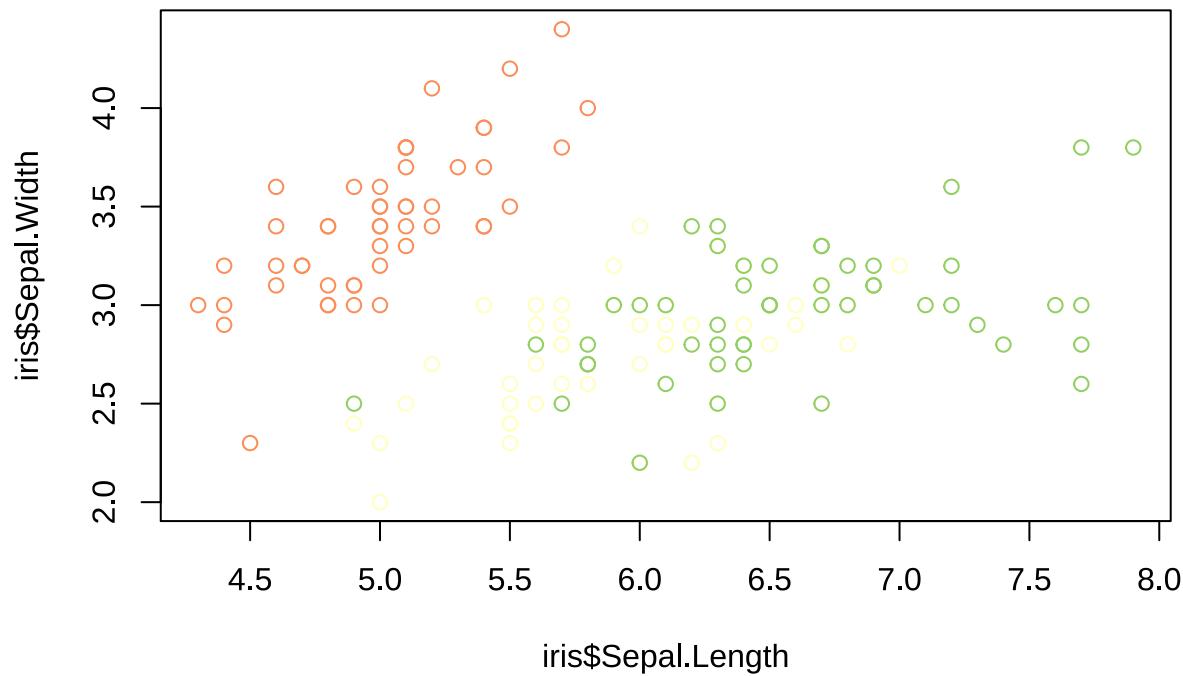
- **连续型 (Sequential)** 指生成一系列连续渐变的颜色，通常用来标记连续型数值的大小。共 18 组颜色，每组分为 9 个渐变颜色展示。
- **极端型 (Diverging)** 指生成用深色强调两端、浅色标示中部的系列颜色，可用来标记数据中的离群点。共 9 组颜色，每组分为 11 个渐变颜色展示。
- **离散型 (Qualitative)** 指生成一系列彼此差异比较明显的颜色，通常用来标记分类数据。共 8 组颜色，每组渐变颜色数不同。

```
par(mfrow = c(1, 3))
library(RColorBrewer)
par(mar = c(0.1, 3, 0.1, 0.1))
```

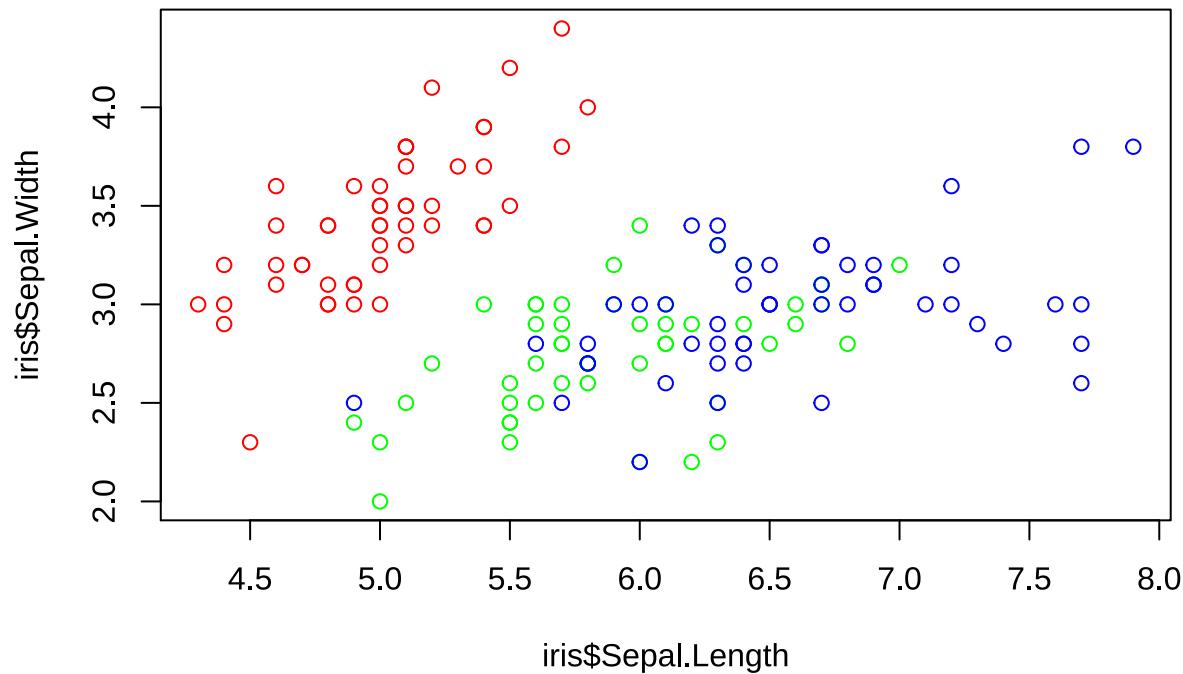
```
display.brewer.all(type = "seq")
display.brewer.all(type = "div")
display.brewer.all(type = "qual")
```



```
library(RColorBrewer)
my_col <- brewer.pal(3, "RdYlGn")
# brewer.pal(n, name), 其中 n 为颜色的数量, name 表示颜色组的名称
plot(iris$Sepal.Length, iris$Sepal.Width, col = rep(my_col, each = 50))
```



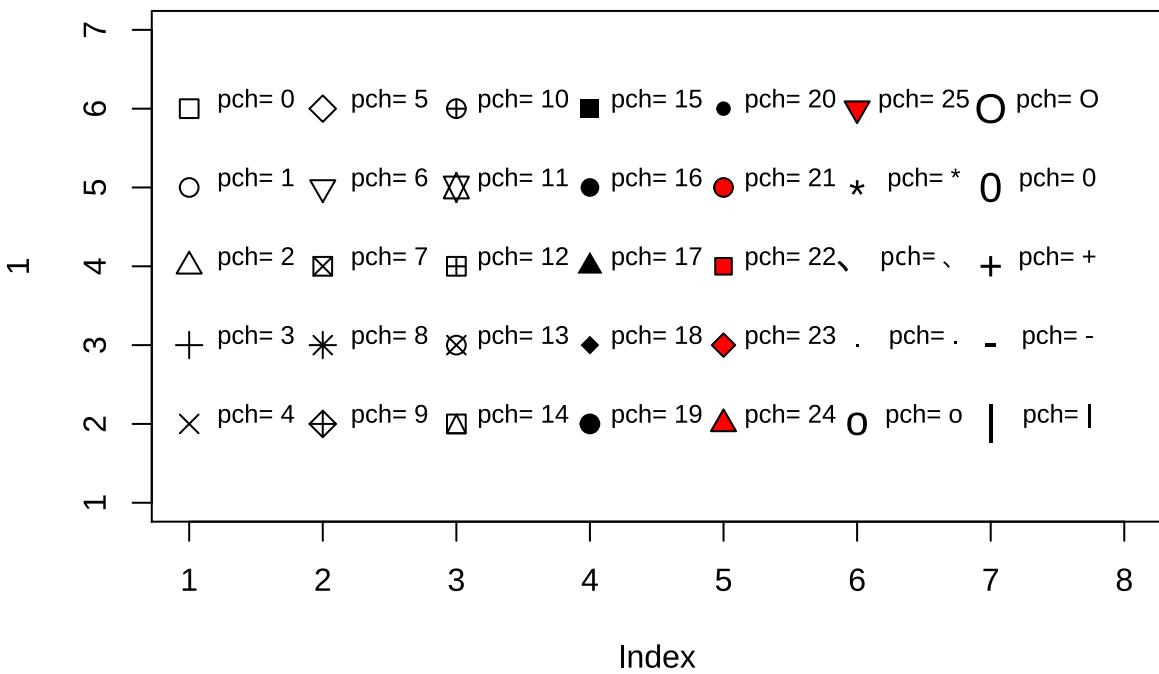
```
plot(iris$Sepal.Length, iris$Sepal.Width, col = rep(rainbow(3), each = 50))
```



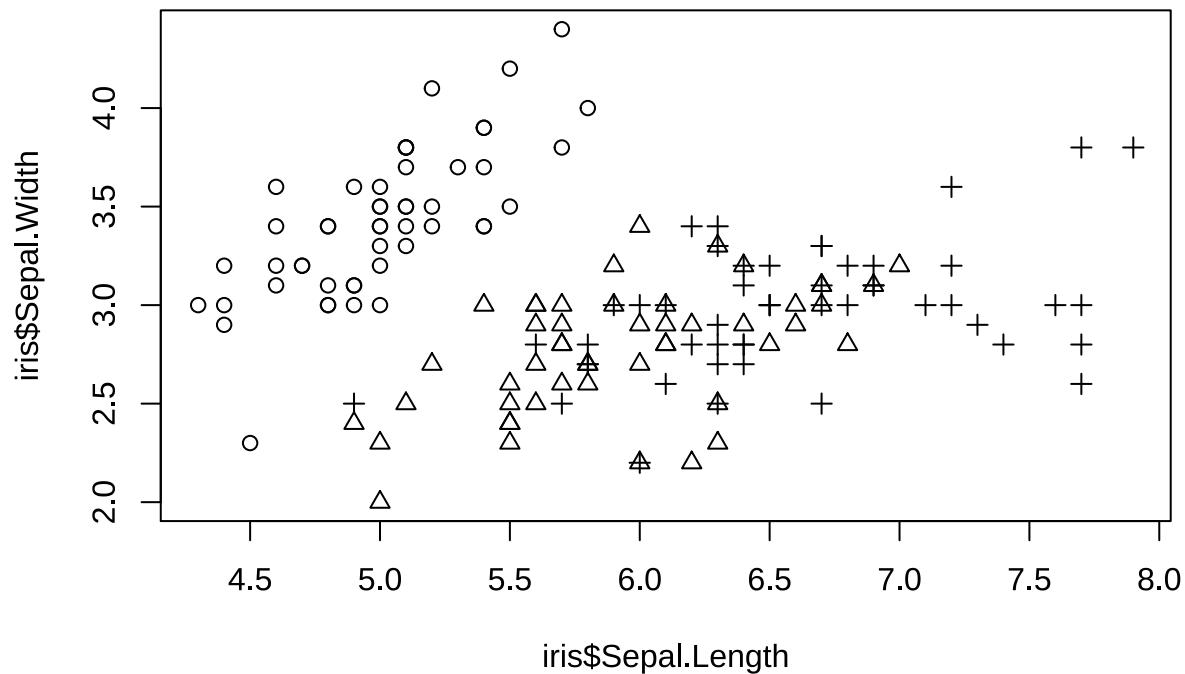
1.2.2 修改点符号与线条

1.2.2.1 点样式

参数	描述
pch	点的样式，取整数0-25或字符"*,"," ","."," o"," O"," 0"," +"," -"," "等
cex	点的大小，1（默认）表示不缩放，小于1表示缩放，大于1表示放大
col	点边框填充的颜色
bg	点内部填充的颜色，仅限21-25样式的点
font	字体设置，1（默认）为正常字体，2表示粗体，3表示斜体，4表示粗斜体
lwd	点边框的宽度，1（默认）表示正常宽度，小于1表示缩放，大于1表示放大



```
plot(iris$Sepal.Length, iris$Sepal.Width, pch = rep(1:3, each = 50))
```



```
# plot(1:10, pch=21, cex=1.5, col='red', bg = 'blue', lwd=5)
```

1.2.2.2 线条样式

R 语言提供了绘制不同类别的线条的多种函数，主要有

- lines: 绘制曲线
- abline: 绘制直线
- segments: 绘制线段
- arrows: 在线段加上箭头
- grid: 绘制网格线

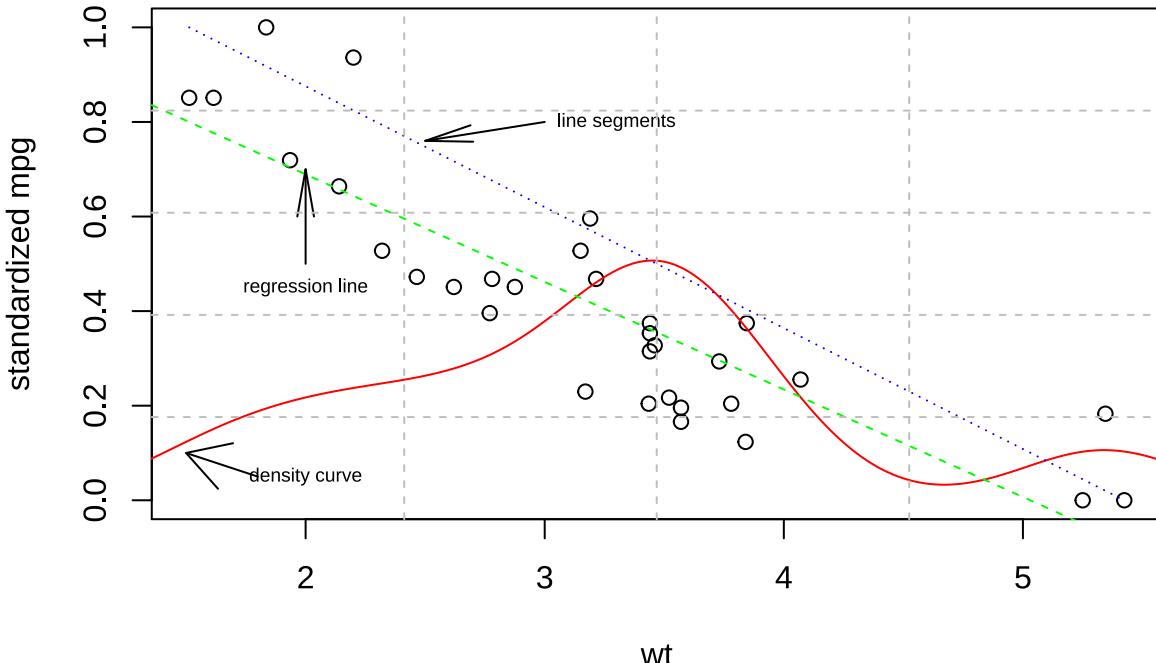
参数	描述
lty	线条样式, 0 表示不画线, 1 表示实线, 2 表示虚线, 3 表示点线。
lwd	线条粗细, 1 (默认) 表示正常宽度, 小于 1 表示缩放, 大于 1 表示放大

以 mtcars 数据集为例来展示实际绘图过程中线条的应用。

```

attach(mtcars)
smpg = (mpg - min(mpg))/(max(mpg) - min(mpg))
plot(wt, smpg, ylab = "standardized mpg")
# 添加核密度曲线图
lines(density(wt), col = "red")
# 指向密度曲线的箭头
arrows(1.8, 0.05, 1.5, 0.1)
text(2, 0.05, "density curve", cex = 0.6)
# 添加回归线
abline(lm(smpg ~ wt), lty = 2, col = "green")
# 指向回归直线的箭头
arrows(2, 0.5, 2, 0.7, angle = 10, cex = 0.5)
text(2, 0.45, "regression line", cex = 0.6)
# wt 与 mpg 反向线性相关, 添加最大最小值线段表现这种关系
segments(min(wt), max(smpg), max(wt), min(smpg), lty = 3, col = "blue")
# 指向最大最小值线段的箭头
arrows(3, 0.8, 2.5, 0.76, angle = 10, cex = 0.5)
text(3.3, 0.8, "line segments", cex = 0.6)
# 添加网格线作为背景
grid(nx = 4, ny = 5, lty = 2, col = "grey")

```



1.2.2.3 修改文本参数

title、text 和 mtext 函数可以在打开的画布上添加文字元素。

- title 可以添加标题元素；
- text 可以任意位置添加文本；
- mtext 函数则是在四条边上添加文本。

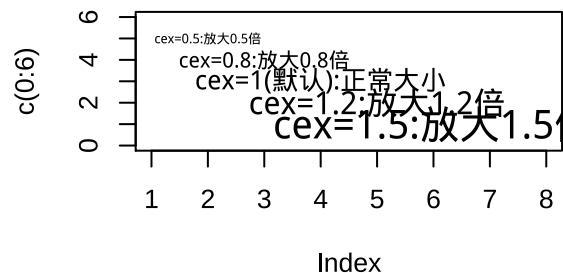
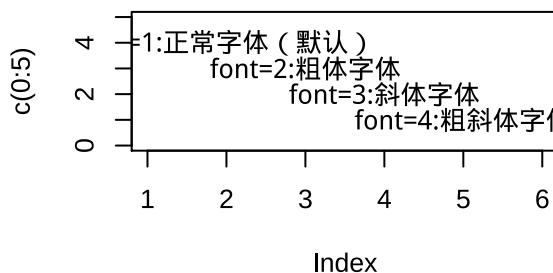
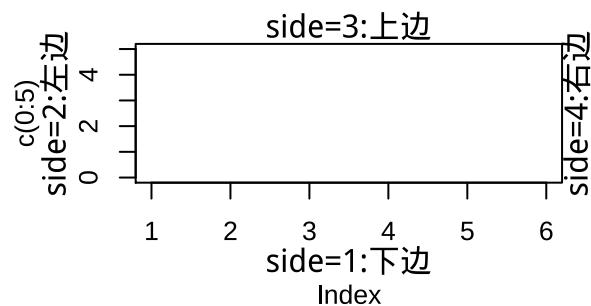
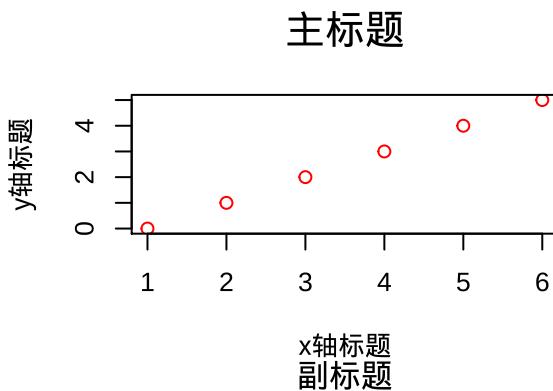
```
par(mfrow = c(2, 2))

# 图一：图形添加标题
plot(c(0:5), col = "red", xlab = "", ylab = "")
title(main = list(" 主标题", cex = 1.5), sub = list(" 副标题", cex = 1.2), xlab = "x 轴标题",
      ylab = "y 轴标题")

# 图二：图形周边添加文本
plot(c(0:5), col = "white")
mtext("side=1: 下边", side = 1, line = 2)
mtext("side=2: 左边", side = 2, line = 2)
mtext("side=3: 上边", side = 3)
mtext("side=4: 右边", side = 4)

# 图三：字体展示
plot(c(0:5), col = "white")
text(2, 4, labels = "font=1: 正常字体 (默认)", font = 1)
text(3, 3, labels = "font=2: 粗体字体", font = 2)
text(4, 2, labels = "font=3: 斜体字体", font = 3)
text(5, 1, labels = "font=4: 粗斜体字体", font = 4)

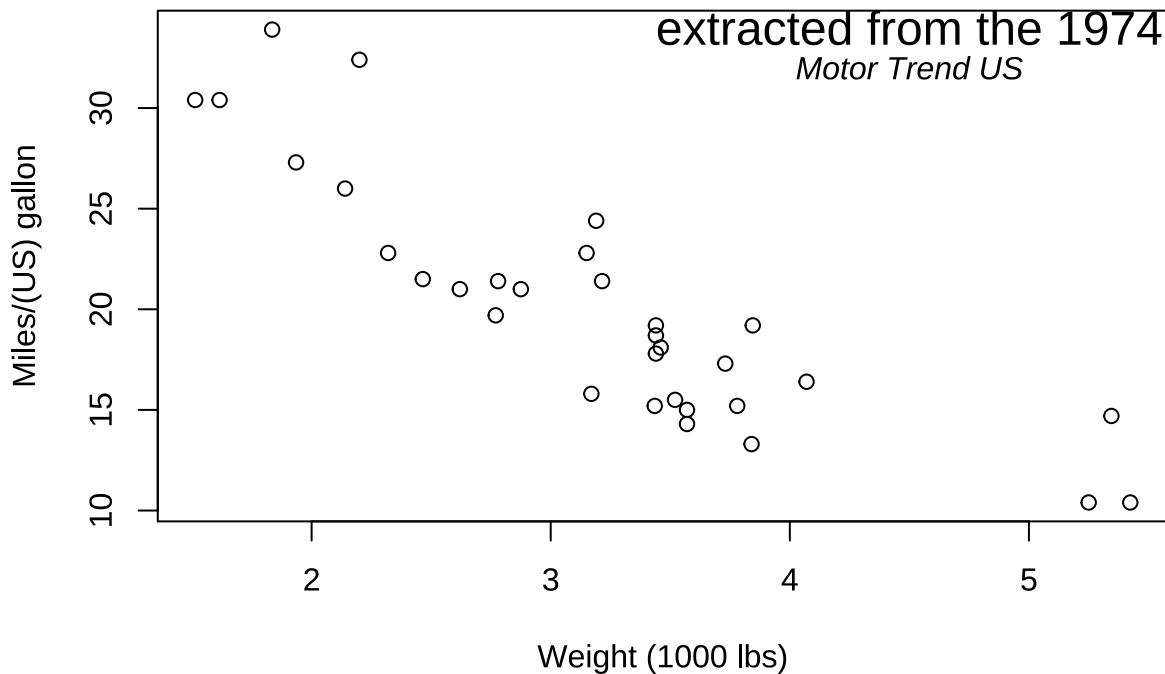
# 图四：字体大小展示
plot(c(0:6), col = "white", xlim = c(1, 8))
text(2, 5, labels = "cex=0.5: 放大 0.5 倍", cex = 0.5)
text(3, 4, labels = "cex=0.8: 放大 0.8 倍", cex = 0.8)
text(4, 3, labels = "cex=1(默认): 正常大小", cex = 1)
text(5, 2, labels = "cex=1.2: 放大 1.2 倍", cex = 1.2)
text(6, 1, labels = "cex=1.5: 放大 1.5 倍", cex = 1.5)
```



例子：

```
attach(mtcars)
plot(wt, mpg, xlab = "Weight (1000 lbs)", ylab = "Miles/(US) gallon") # 绘图，并修改 x, y 轴的标题
title(main = list("mtcars wt V.S. mpg", cex = 1.5)) # 添加标题
text(4.5, 34, labels = "extracted from the 1974", cex = 1.5) # 说明数据来源
text(4.5, 32, labels = "Motor Trend US", font = 3) # 杂志名称
```

mtcars wt V.S. mpg



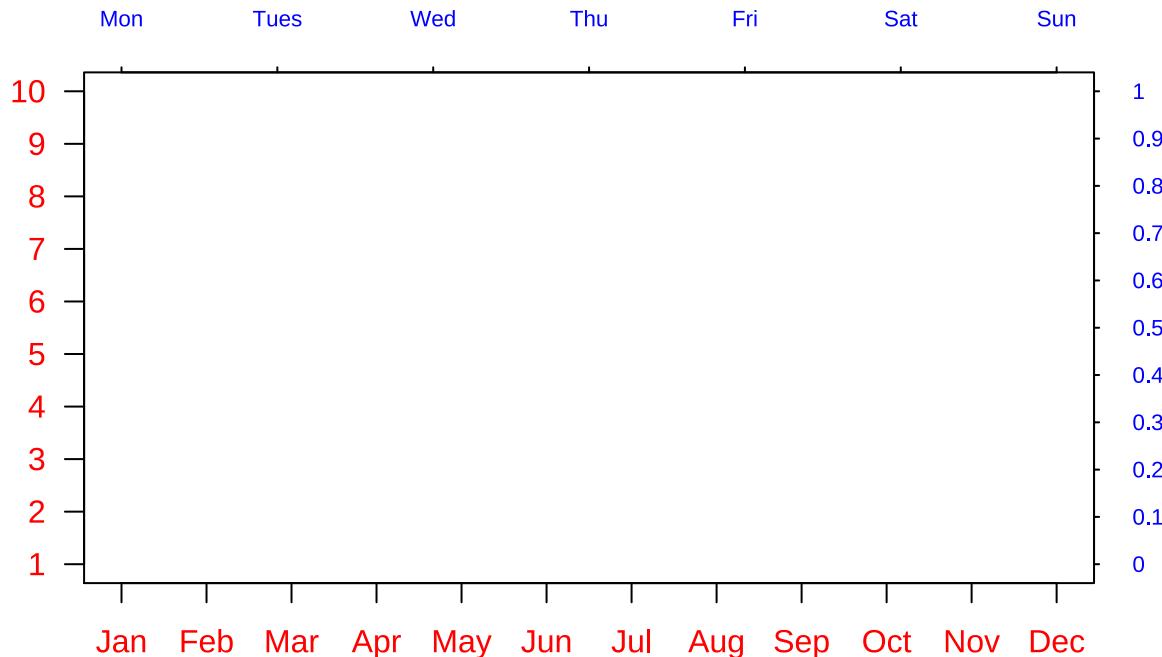
1.2.2.4 设置坐标轴

使用 `axis()` 进行设置坐标轴。

参数	描述
<code>axes</code>	逻辑参数，如果 <code>axes=TRUE</code> （默认），则显示坐标轴，如果 <code>axes=FALSE</code> ，则隐藏坐标轴。
<code>xaxt/yaxt</code>	坐标轴样式，默认值“ <code>s</code> ”，表示 <code>x/y</code> 轴以标准样式显示，取值“ <code>n</code> ”表示隐藏 <code>x/y</code> 轴
<code>xaxs/yaxs</code>	坐标轴计算方式，默认值“ <code>r</code> ”表示把原始数据的范围向外扩大4%，作为 <code>x/y</code> 轴范围，取值“ <code>l</code> ”表示 <code>x/y</code> 轴范围为原始数据范围
<code>xlim/ylim</code>	坐标轴范围，设置为 <code>c(from,to)</code> ， <code>from</code> 是 <code>x/y</code> 轴的首坐标， <code>to</code> 是尾坐标

```
plot(c(1:12), col = "white", xaxt = "n", yaxt = "n", ann = FALSE)
axis(1, at = 1:12, col.axis = "red", labels = month.abb)
axis(2, at = seq(1, 12, length = 10), col.axis = "red", labels = 1:10, las = 2)
```

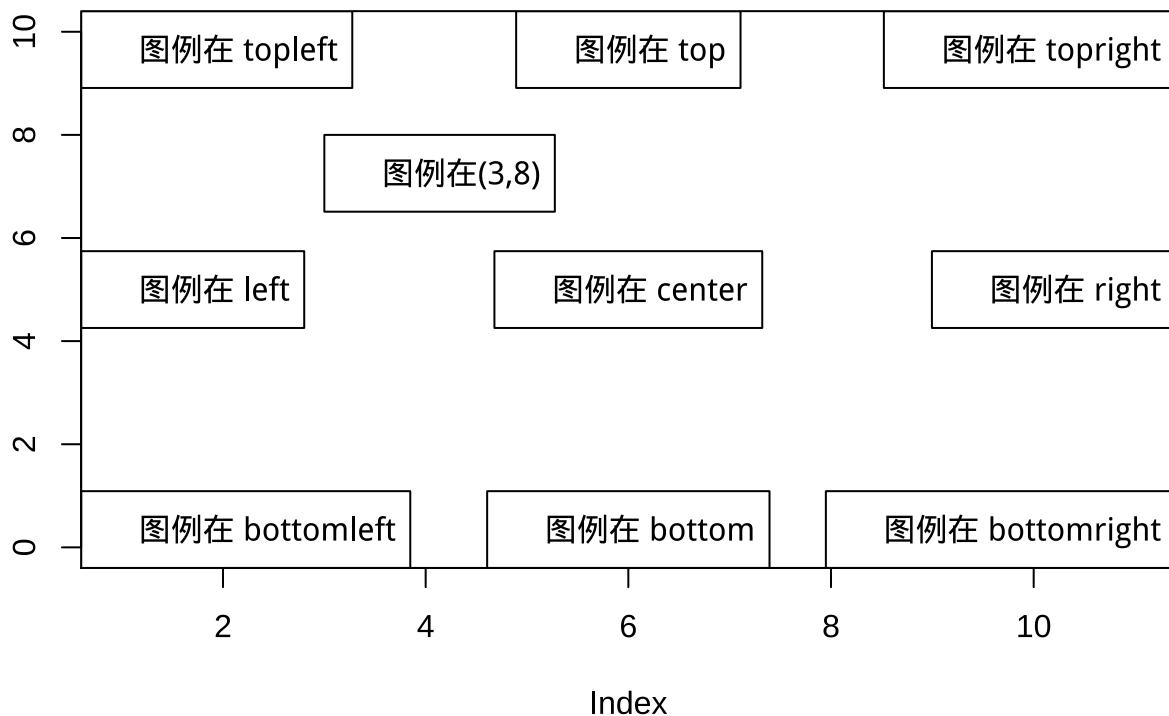
```
axis(3, at = seq(1, 12, length = 7), col.axis = "blue", cex.axis = 0.7, tck = -0.01,
  labels = c("Mon", "Tues", "Wed", "Thu", "Fri", "Sat", "Sun"))
axis(4, at = seq(1, 12, length = 11), col.axis = "blue", cex.axis = 0.7, tck = -0.01,
  labels = seq(0, 1, 0.1), las = 2)
```



1.2.2.5 添加图例

legend 函数的绘制图例的位置效果

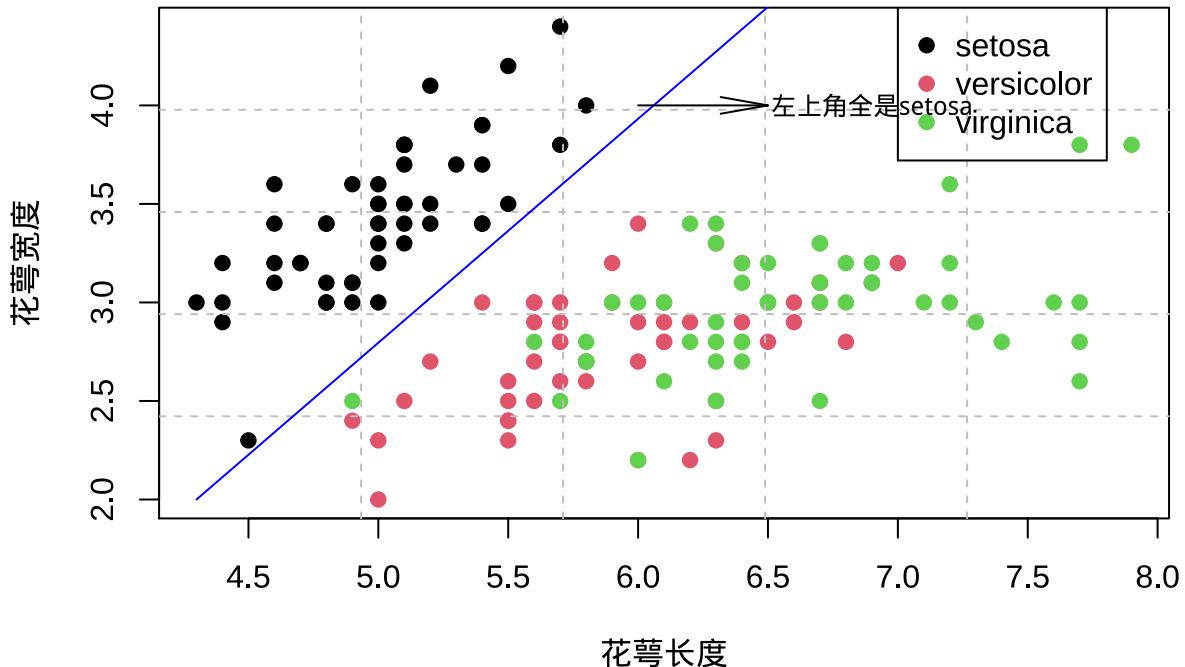
```
local = c("bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright",
  "right", "center")
par(mar = c(4, 2, 4, 2), pty = "m")
plot(c(0:10), col = "white")
legend(3, 8, " 图例在 (3,8)")
for (i in 1:9) {
  legend(local[i], paste(" 图例在", local[i]))
}
```



综合测试：

```
plot(iris$Sepal.Length, iris$Sepal.Width, col = iris$Species, main = list(" 鸢尾花的花萼长与宽的散点图",
  cex = 1.5), xlab = " 花萼长度", ylab = " 花萼宽度", pch = 19)
grid(nx = 5, ny = 5, lty = 2, col = "grey") # 添加网格线
legend(7, 4.5, c("setosa", "versicolor", "virginica"), pch = 19, col = 1:3) # 添加图例
lines(c(4.3, 6.5), c(2, 4.5), col = "blue") # 添加直线
arrows(6, 4, 6.5, 4, angle = 10, cex = 0.5) # 添加箭头
text(6.9, 4, " 左上角全是 setosa", cex = 0.8) # 添加文字说明
```

鸢尾花的花萼长与宽的散点图

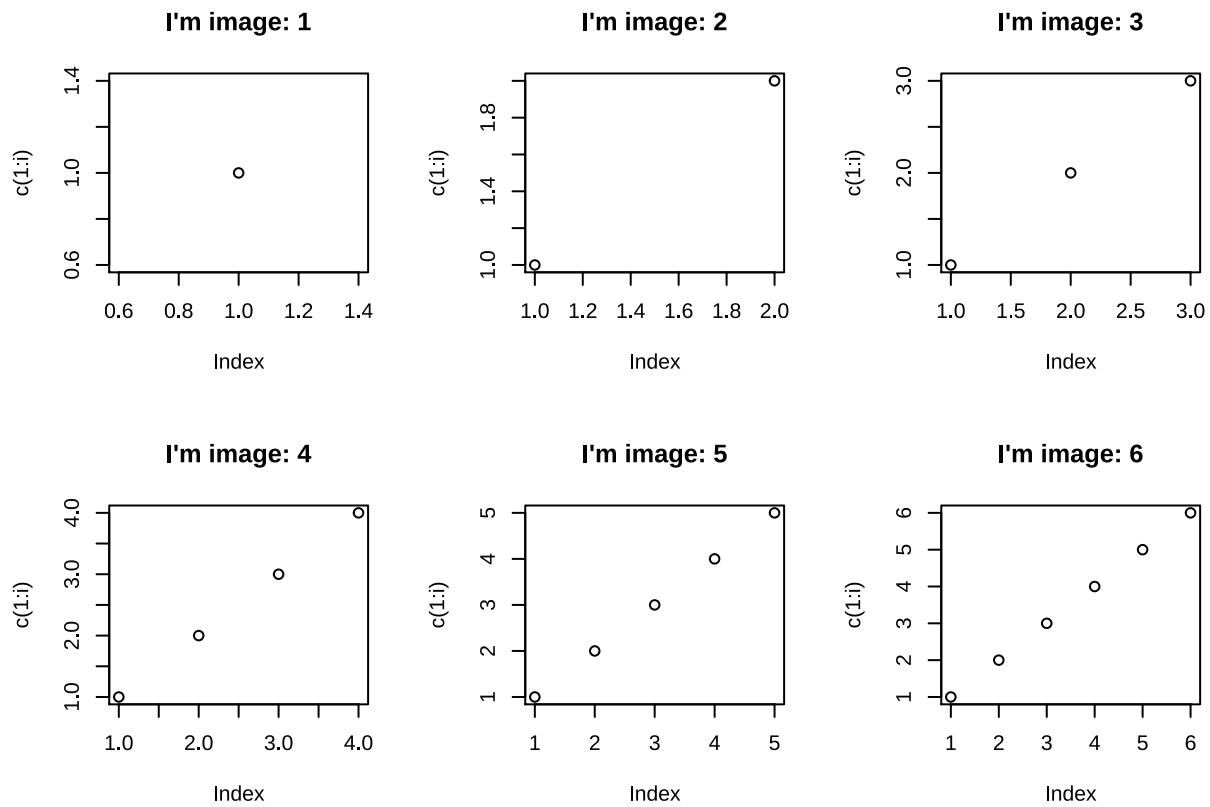


1.3 绘制组合图形

1.3.1 par()

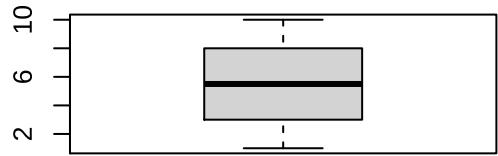
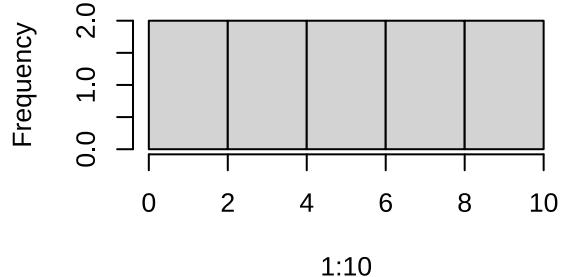
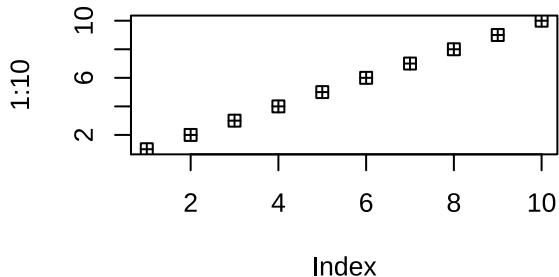
一页多图用 `mfrow` 参数或 `mfcol` 参数规定。

```
mfrow1 = par(mfrow = c(2, 3)) #mar=c(2,2,2,2)
for (i in 1:6) {
  plot(c(1:i), main = paste("I'm image:", i))
}
```



```
par(mfrow=c(2, 2))

op = par(mfrow = c(2, 2))
plot(1:10, pch = 12)
hist(1:10)
boxplot(1:10)
pie(1:10)
```

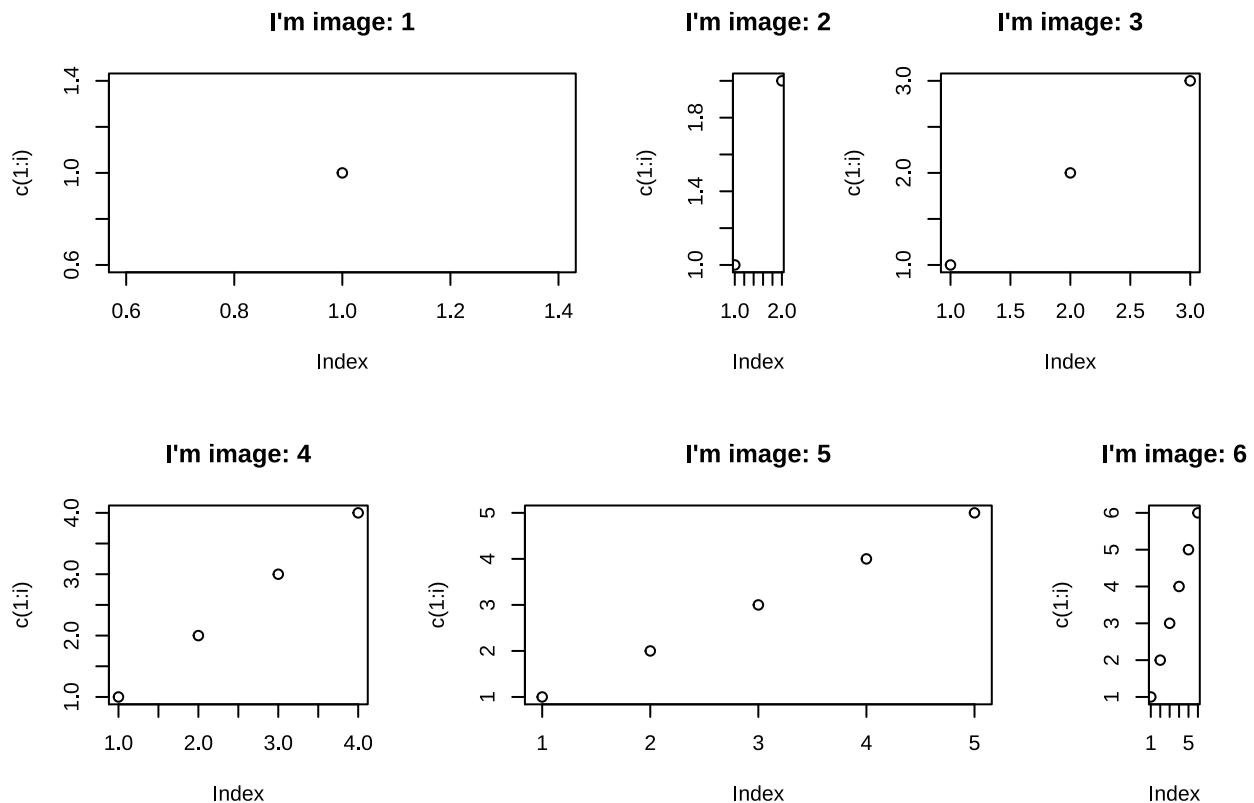


```
par(op)
```

1.3.2 layout

与 par 函数均分画布不同，layout 函数可以不均等的分隔页面

```
mat <- matrix(c(1, 1, 1, 2, 3, 3, 4, 4, 5, 5, 5, 6), nrow = 2, byrow = TRUE)
layout(mat)
for (i in 1:6) {
  plot(c(1:i), main = paste("I'm image:", i))
}
```



1.4 保存图形

1.4.1 使用代码

对于其他格式输出类似 pdf 的输出。

```
pdf("test/2.pdf") # 保存到当前工作目录下
plot(1:10)
dev.off()

## pdf
## 2
```

1.4.2 在 Rstudio 窗口点击按钮保存

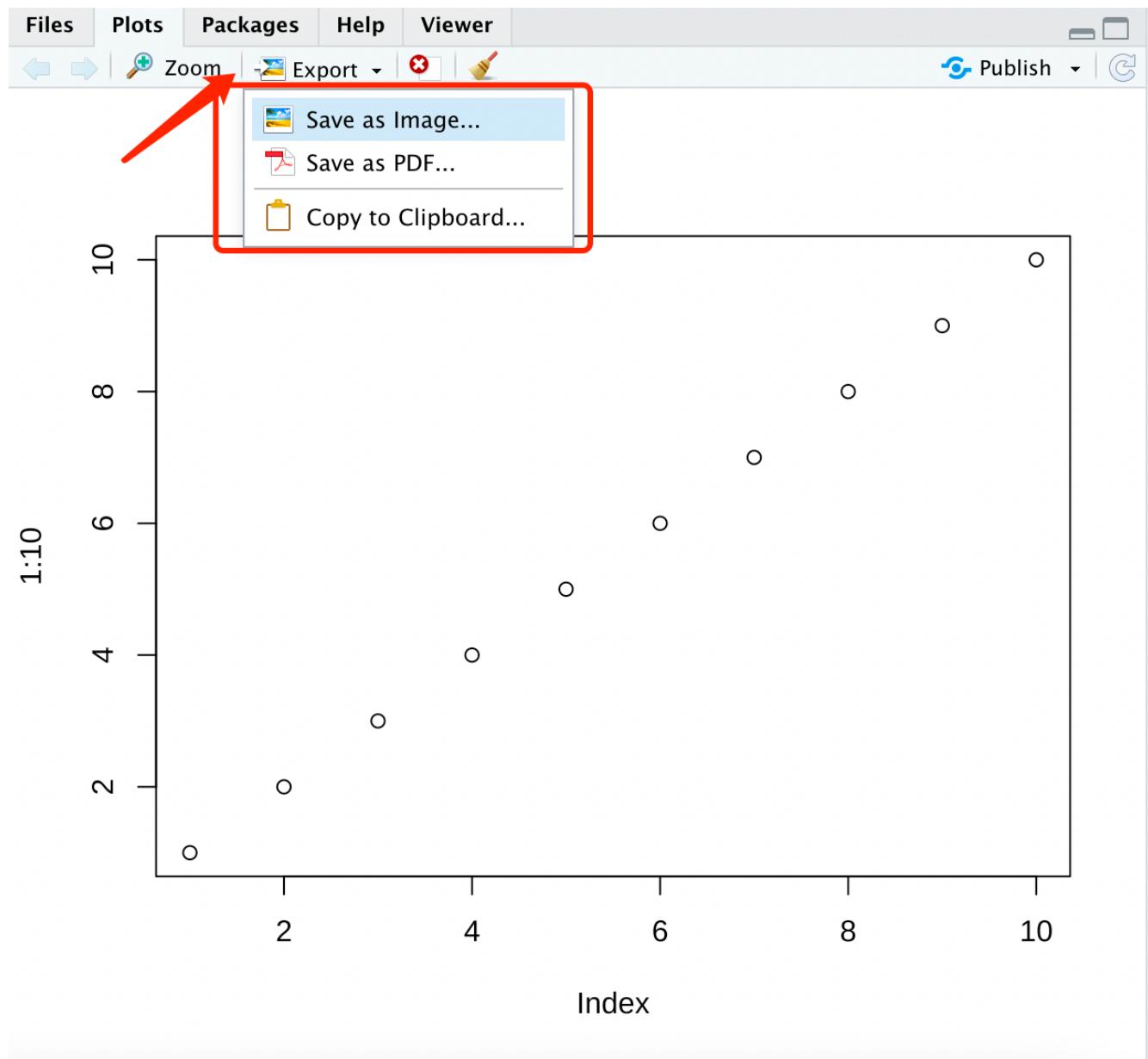


图 1.1: Rstudio 界面右下角



图 1.2: 自定义设置

Chapter 2

使用 `ggplot2` 包绘图

2.1 简介

`ggplot2` 包是 Harley Wickham 在 2005 年创建的，是包含了一套全面而连贯的语法的绘图系统。

弥补了 R 中创建图形缺乏一致性的缺点，且不会局限于一些已经定义好的统计图形，可以根据需要创造出任何有助于解决所遇到问题的图形。

核心理念：将绘图与数据分离，数据相关的绘图与数据无关的绘图分离，按图层作图。

2.2 `qplot`

`ggplot2` 包的绘图语言与常用的绘图函数的使用方法不同，为了让读者快速使用 `ggplot2` 包，包的作者 Harley Wickham 提供了 `qplot` 函数（quick plot），让人在了解 `ggplot2` 的语言逻辑之前，就能迅速实现数据的可视化。

鸢尾花数据集 `iris`

```
head(iris, 10)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa
## 7	4.6	3.4	1.4	0.3	setosa



图 2.1: Harley Wickham

```

## 8      5.0      3.4      1.5      0.2  setosa
## 9      4.4      2.9      1.4      0.2  setosa
## 10     4.9      3.1      1.5      0.1  setosa

str(iris)

## 'data.frame':   150 obs. of  5 variables:
## $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species     : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...

```

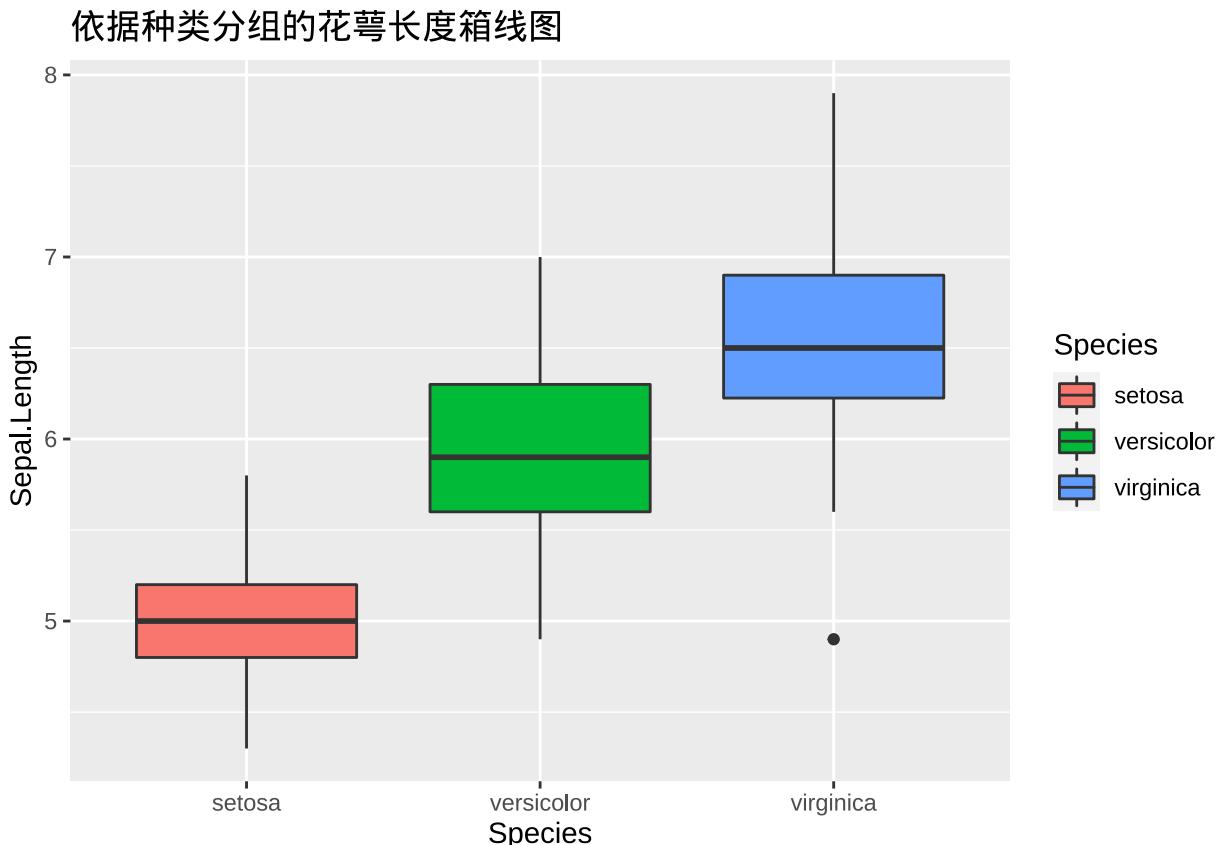
- 例子一:

创建一个以物种种类为分组的花萼长度的箱线图，箱线图的颜色依据不同的物种种类而变化。

```

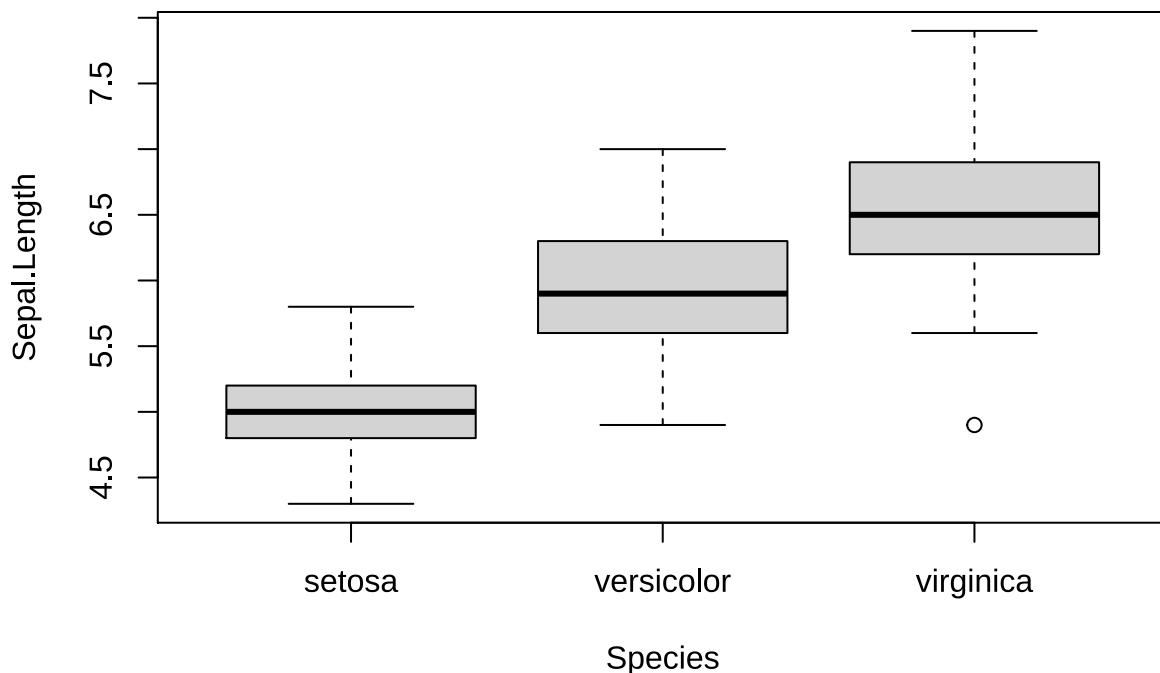
library(ggplot2)
qplot(Species, Sepal.Length, data = iris, geom = "boxplot", fill = Species, main = " 依据种类分组的花萼长度箱线图"

```



```
boxplot(Sepal.Length ~ Species, data = iris, main = " 依据种类分组的花萼长度箱线图")
```

依据种类分组的花萼长度箱线图

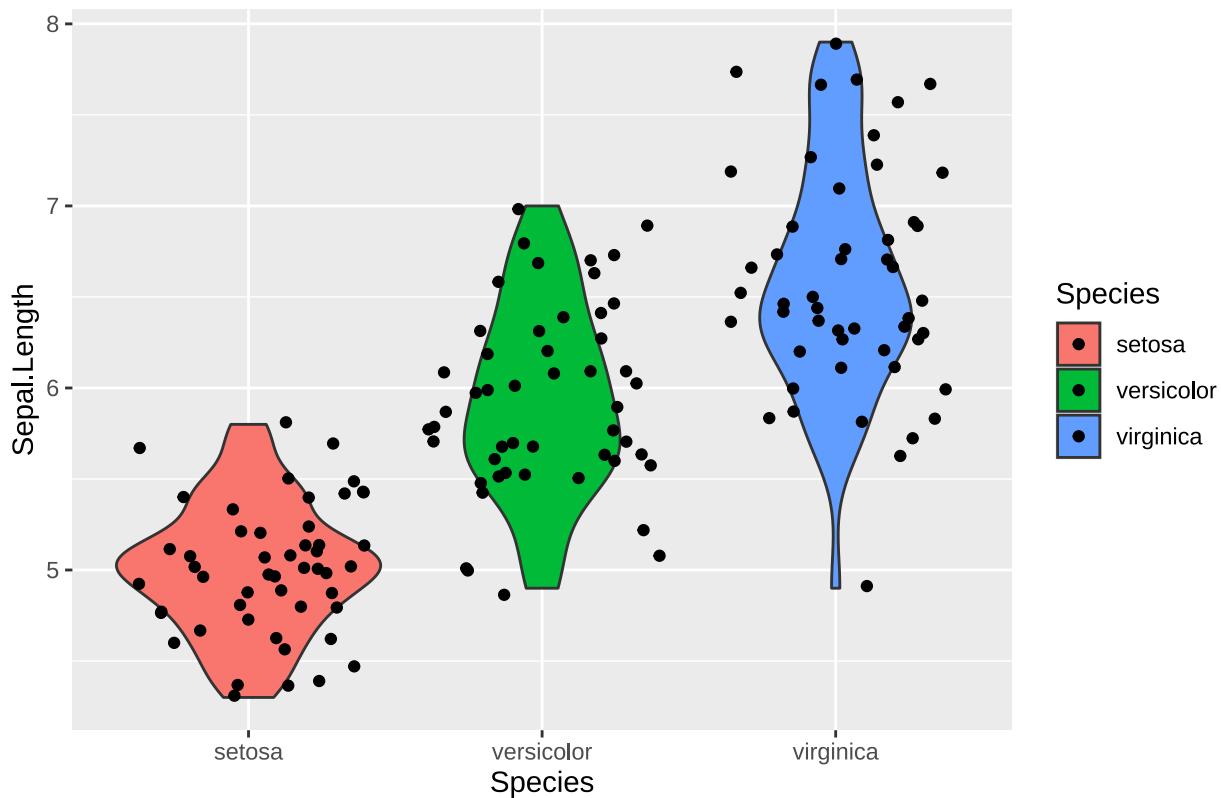


- 例子二：

利用 qplot 函数画出小提琴图，只需要将 geom 设置为 “violin” 即可，并添加扰动以减少数据重叠。

```
qplot(Species, Sepal.Length, data = iris, geom = c("violin", "jitter"), fill = Species,  
main = " 依据种类分组的小提琴图")
```

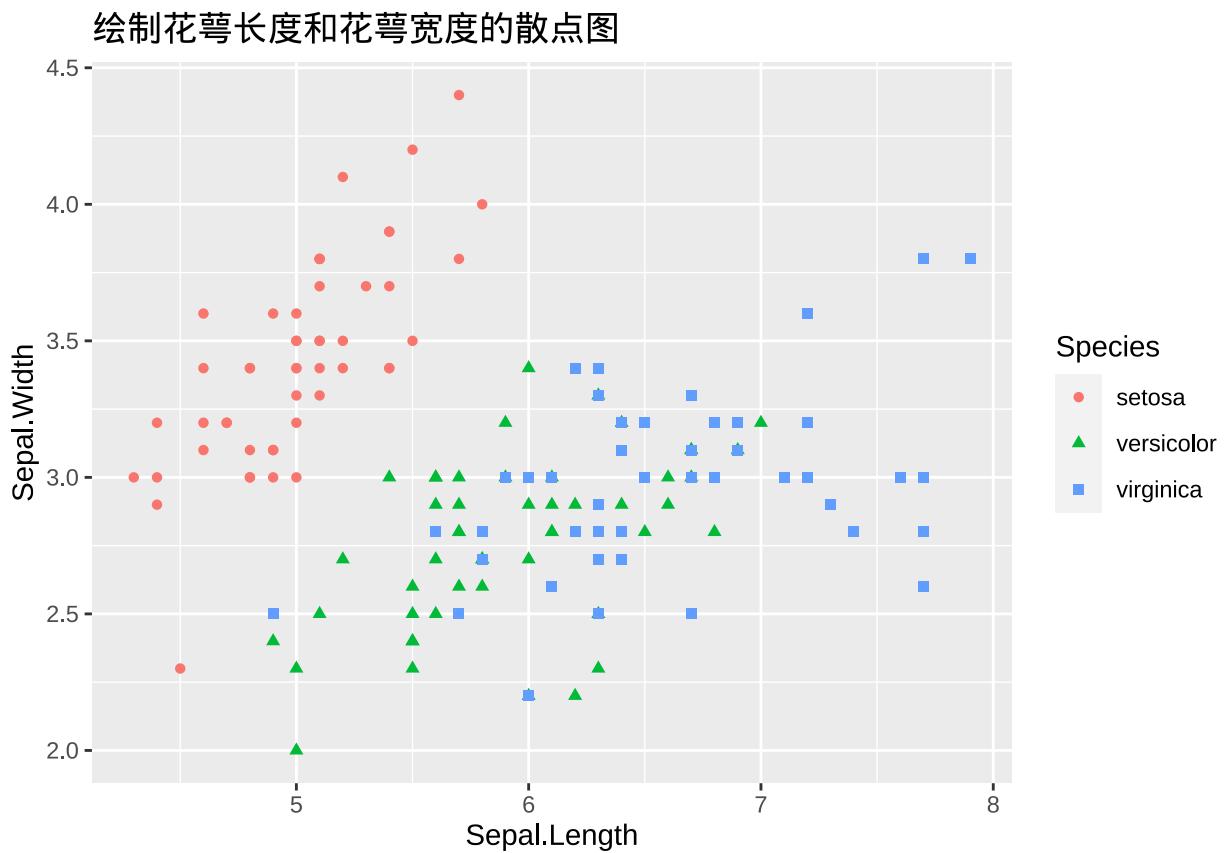
依据种类分组的花萼长度小提琴图



- 例子三：

建一个以花萼长度和花萼宽度的散点图，并利用颜色和符号形状区分物种种类。

```
qplot(Sepal.Length, Sepal.Width, geom = "point", data = iris, colour = Species, shape = Species,
      main = "绘制花萼长度和花萼宽度的散点图")
```

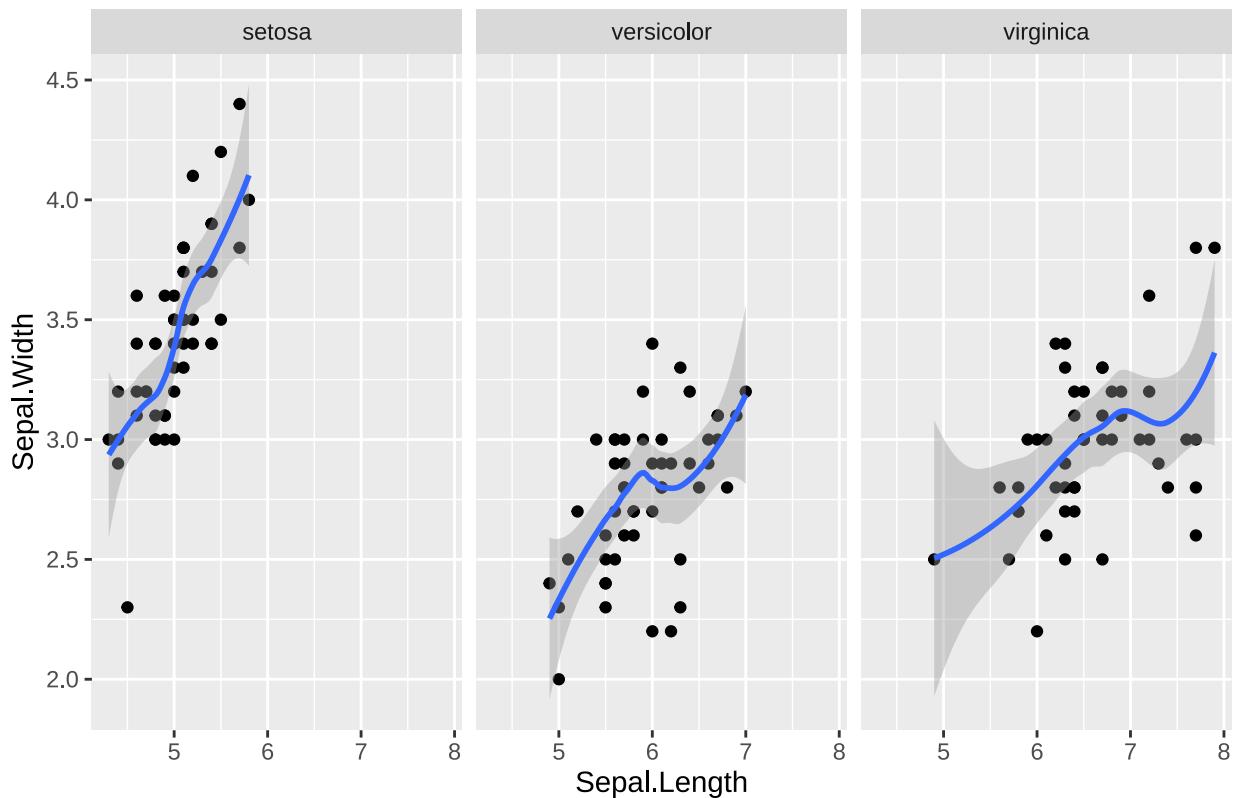


- 例子四：

利用 facets 参数绘制分面板散点图，并增加光滑曲线。

```
qplot(Sepal.Length, Sepal.Width, data = iris, geom = c("point", "smooth"),
      facets = ~Species,
      main = "绘制分面板的散点图")
```

绘制分面板的散点图



2.3 ggplot2 包图形语法

推荐书籍：

ggplot2: Elegant Graphics for Data Analysis <https://ggplot2-book.org/>

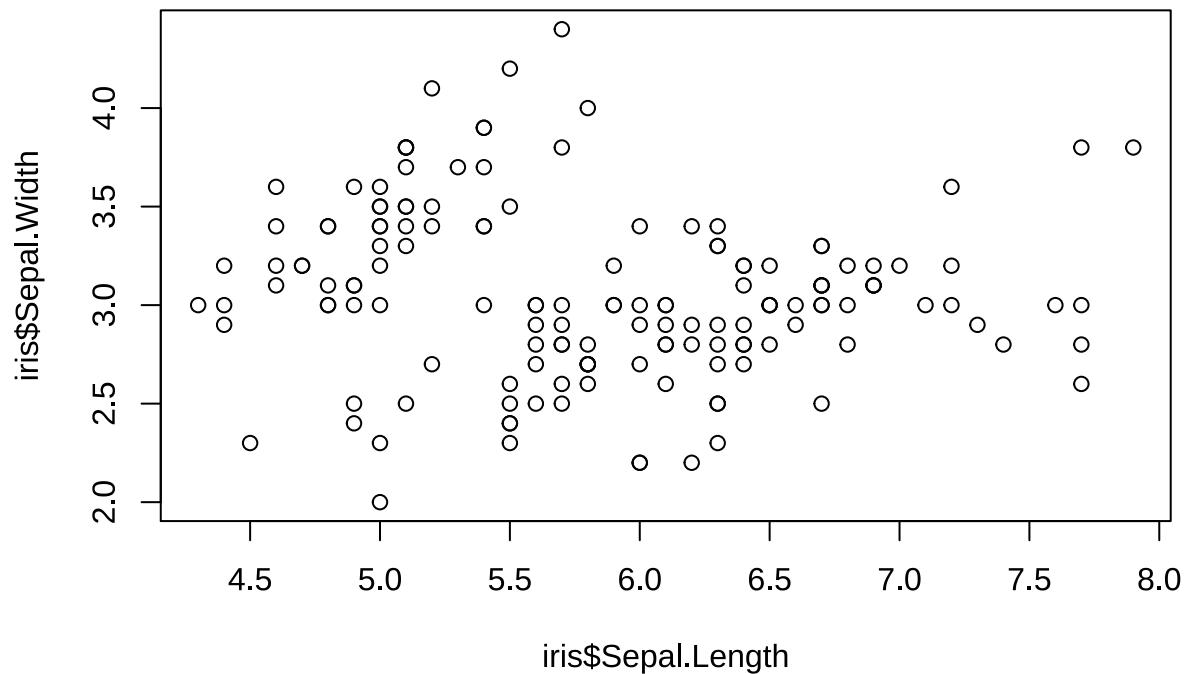
Fundamentals of Data Visualization <https://clauswilke.com/dataviz/>

2.3.1 对比不同画图语法

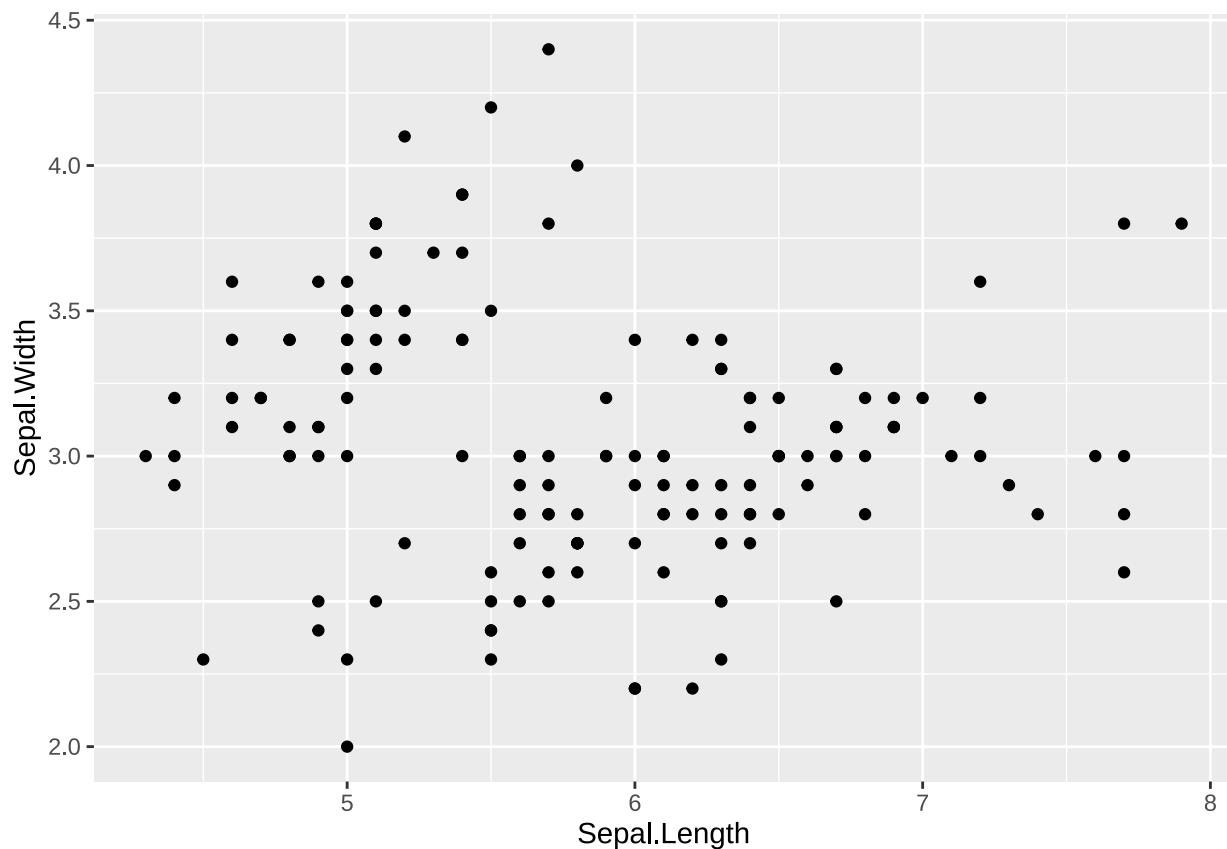
以绘制 iris 数据集中 Sepal.Length 与 Sepal.Width 的散点图为例，分别采用内置的 plot 函数与 ggplot2 包的 ggplot 函数绘制散点图，对比理解 ggplot2 包的语言逻辑。

代码（三种类型）：

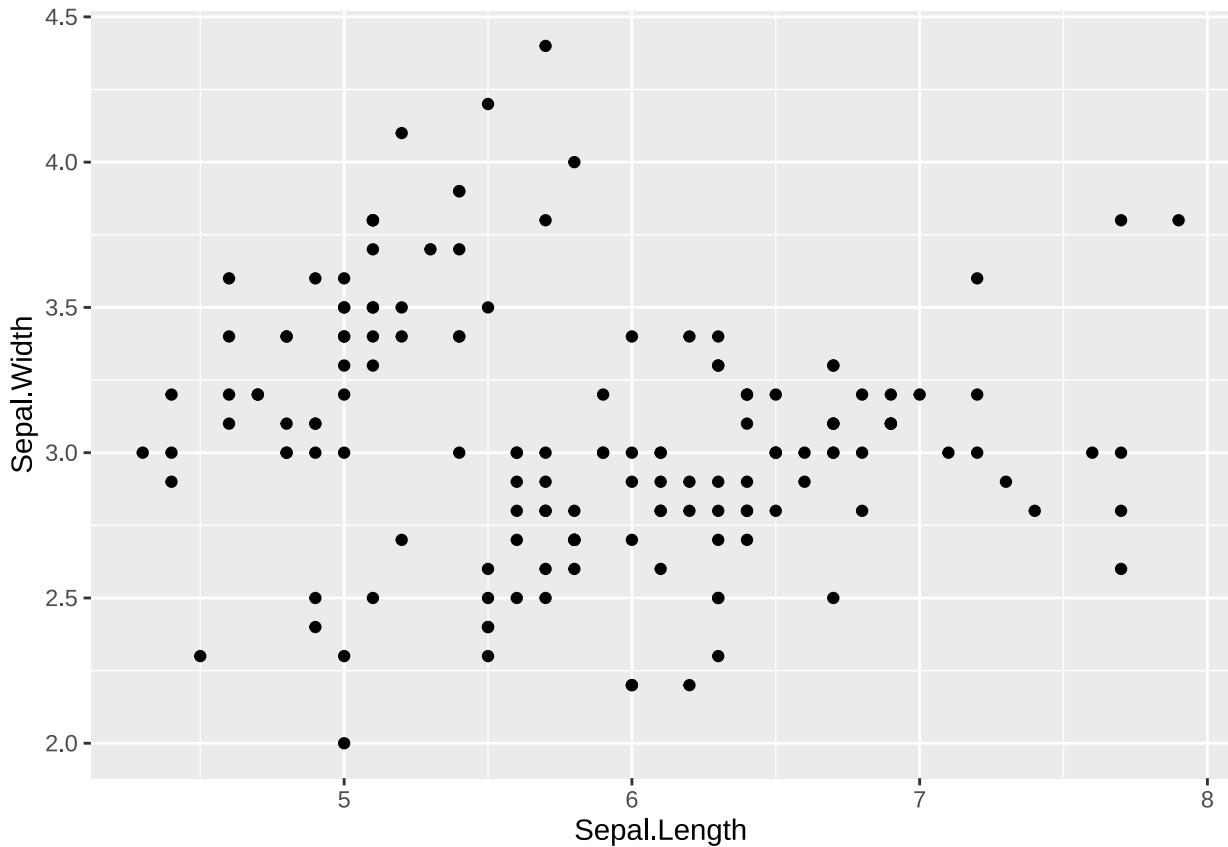
```
# 基础包
plot(iris$Sepal.Length, iris$Sepal.Width)
```



```
# qplot()  
qplot(x = Sepal.Length, y = Sepal.Width, data = iris, geom = "point")
```



```
# ggplot()
ggplot(data= iris, aes(x = Sepal.Length, y = Sepal.Width)) + # 绘制底层画布
geom_point() # 在画布上添加点
```



2.3.2 思想介绍

注：该部分主要参考数据科学中的 R 语言——王敏杰。

ggplot 的绘图有以下几个特点。

1. 有明确的起始（以 ggplot 函数开始）与终止（一句语句一幅图）。
2. ggplot2 语句可以理解为一句语句绘制一幅图，然后进行图层叠加，而叠加是通过”+“号把绘图语句拼接实现的。

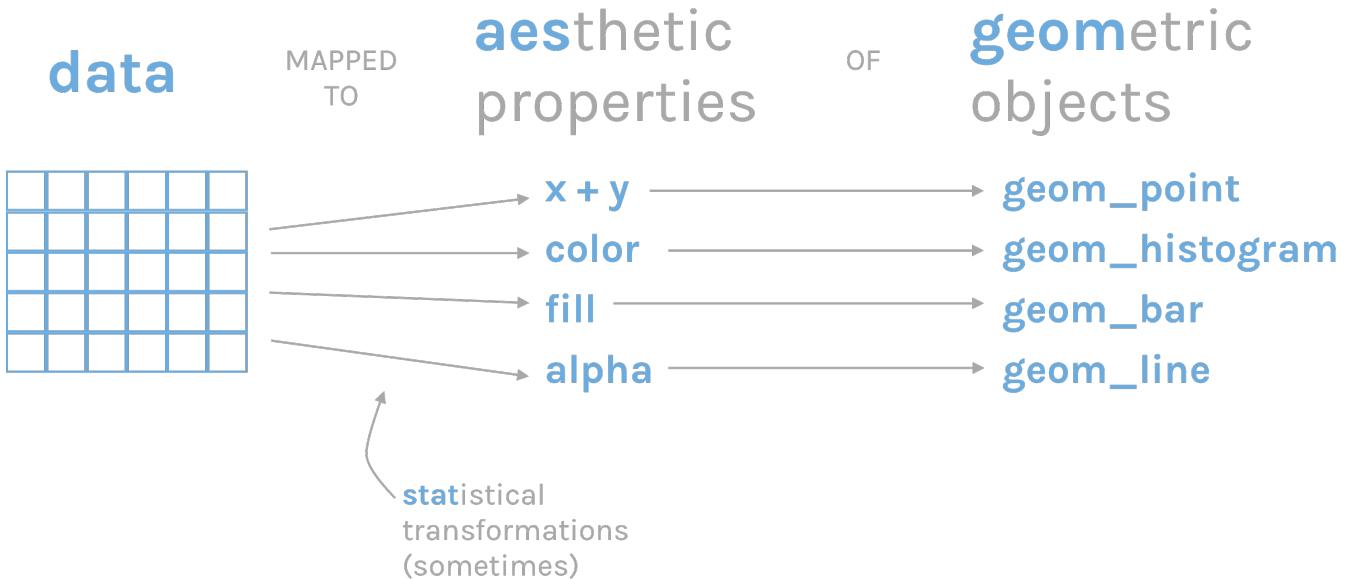
ggplot 函数包括 9 个部件：

- 数据 (data) （数据框）
- 映射 (mapping)
- 几何对象 (geom_point(), geom_boxplot())
- 统计变换 (stats)

- 标度 (scale)
- 坐标系 (coord)
- 分面 (facet)
- 主题 (theme)
- 存储和输出 (output)

其中前三个是必需的。

Hadley Wickham 将这套可视化语法诠释为: 一张统计图形就是从数据到几何对象 (geometric object, 缩写 geom) 的图形属性 (aesthetic attribute, 缩写 aes) 的一个映射。



此外，图形中还可能包含数据的统计变换 (statistical transformation, 缩写 stats)，最后绘制在某个特定的坐标系 (coordinate system, 缩写 coord) 中，而分面 (facet) 则可以用来生成数据不同子集的图形。

例子 (带你入门)

```
ggplot(data = iris, mapping = aes(Petal.Length, Petal.Width)) + geom_point(size = 2,
  alpha = 0.5, col = "red") + geom_smooth(method = "lm", se = F)
```

```
ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +
  <GEOM_FUNCTION>(
    stat = <STAT>,
    position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

图 2.2: 语法模板

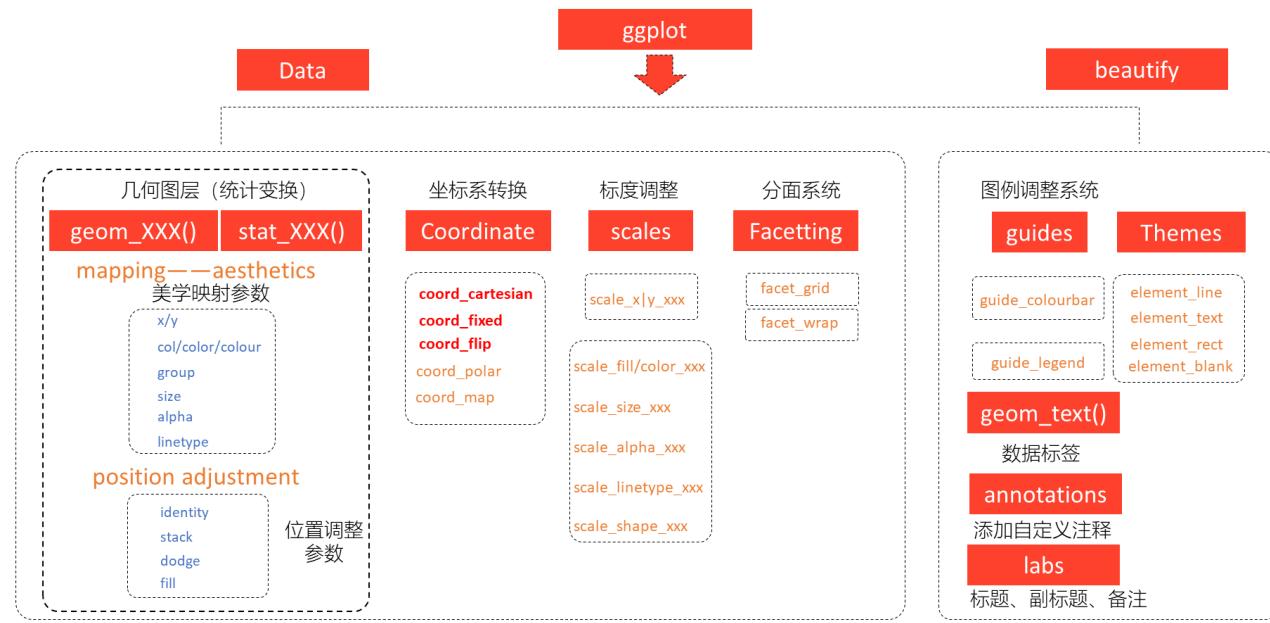
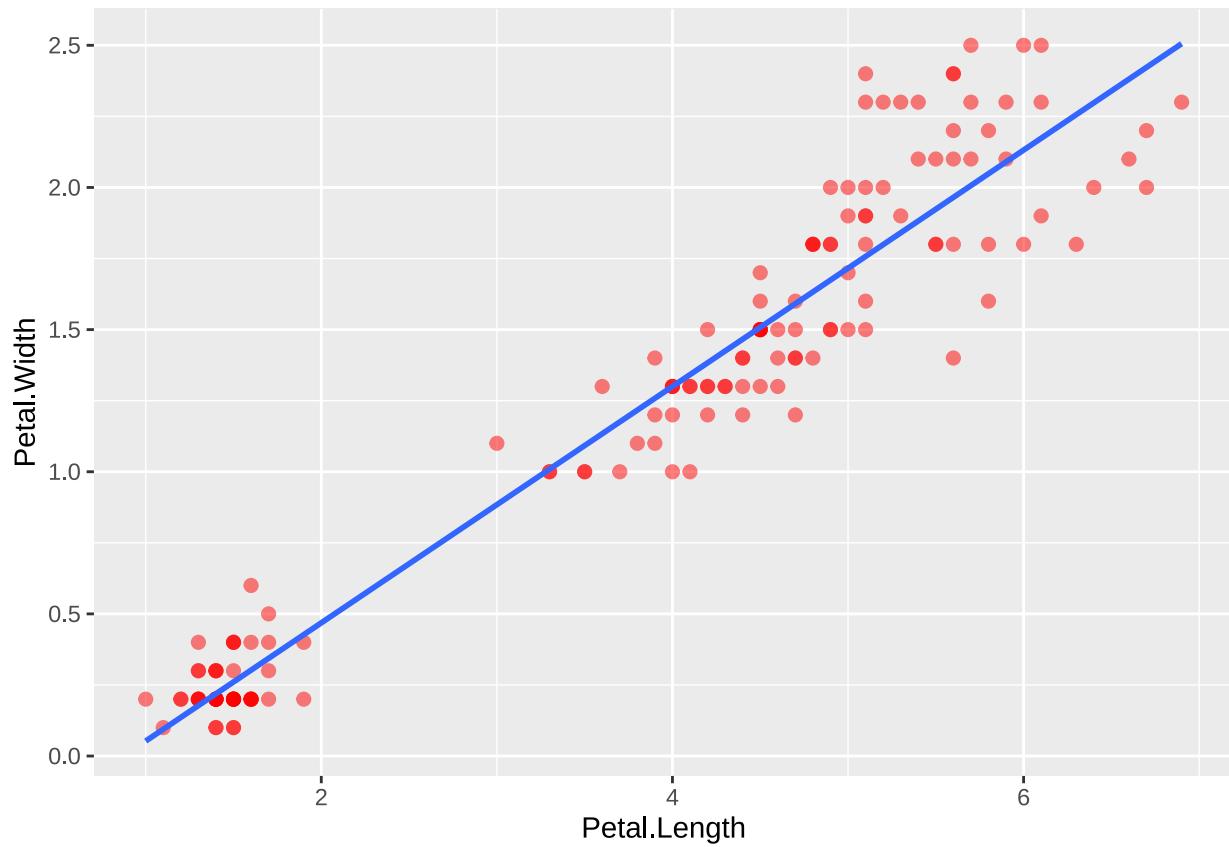
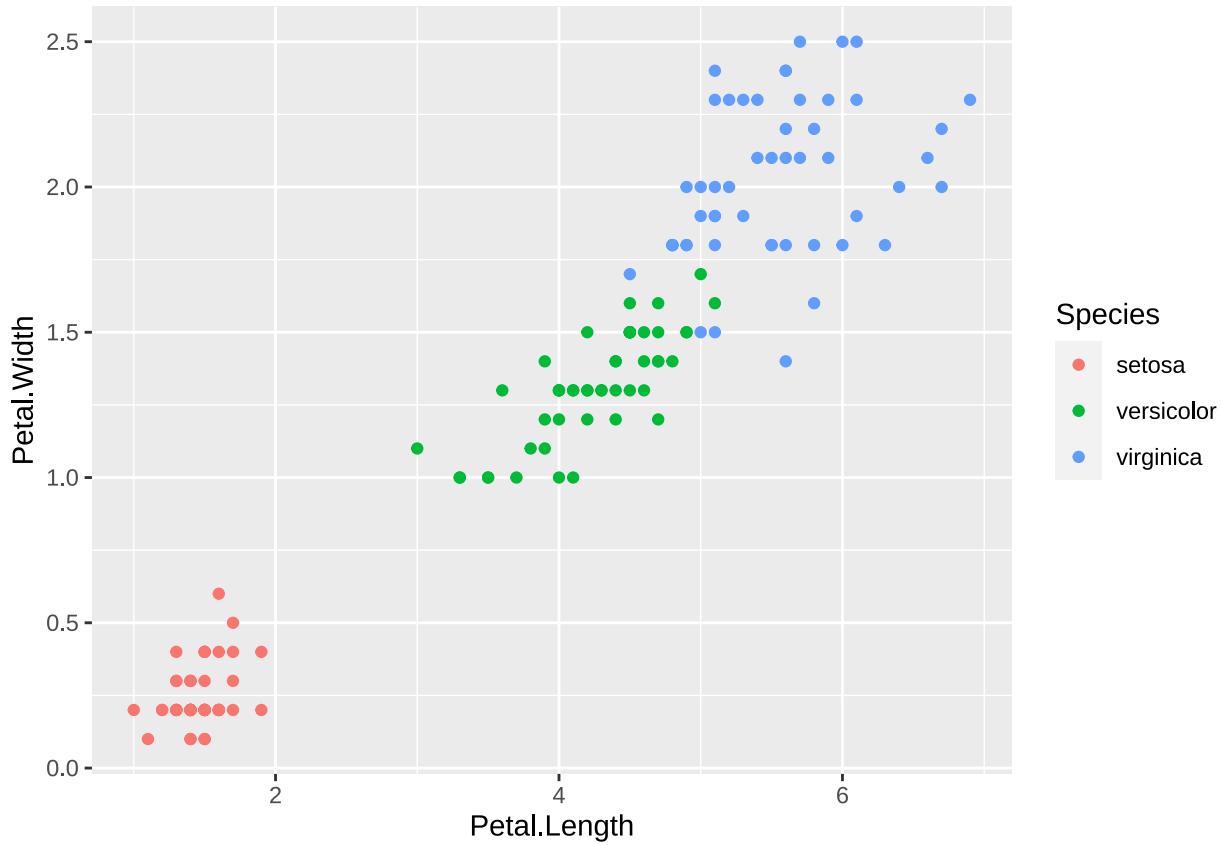


图 2.3: 把这看懂其实差不多了

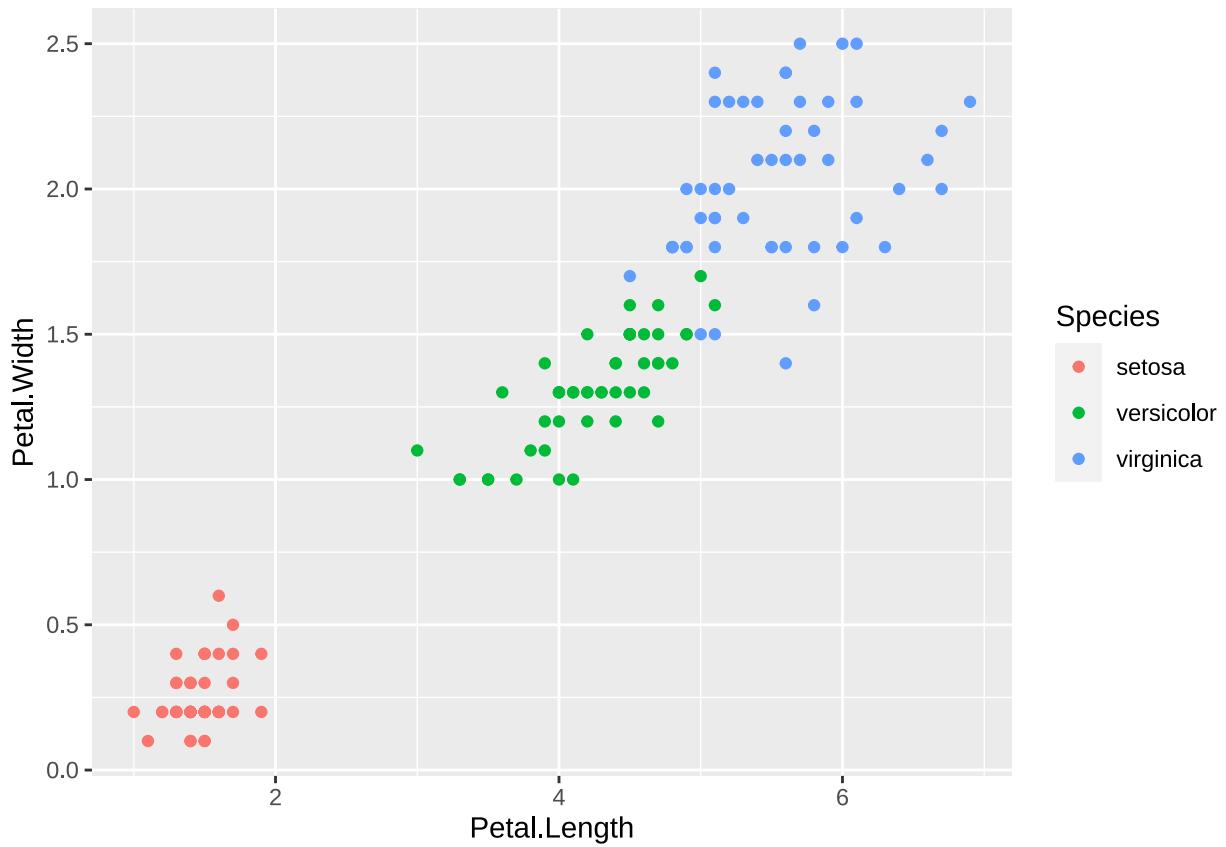


```
ggplot(data = iris, mapping = aes(x = Petal.Length, y = Petal.Width, col = Species)) +  
  geom_point()
```

2.3.3 全局变量 vs. 局部变量



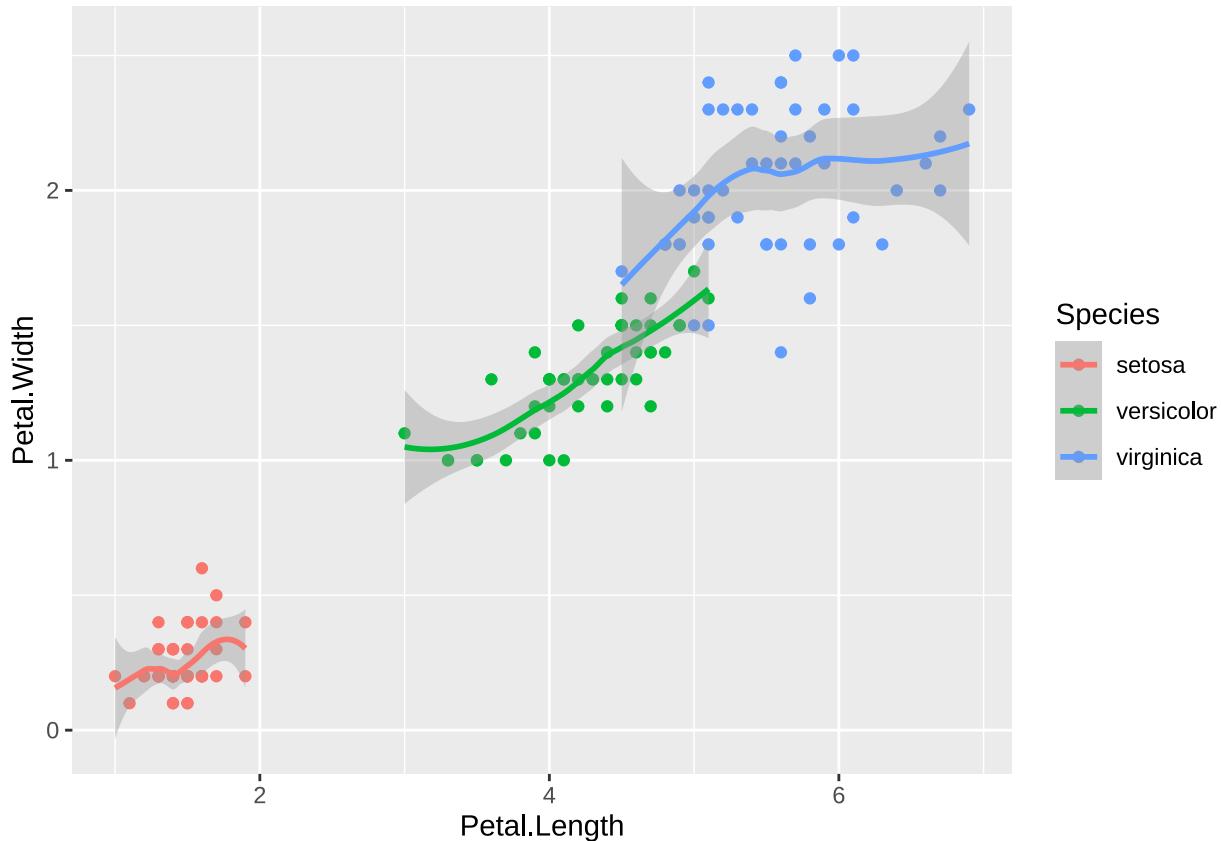
```
ggplot(data = iris) + geom_point(mapping = aes(x = Petal.Length, y = Petal.Width,  
      col = Species))
```



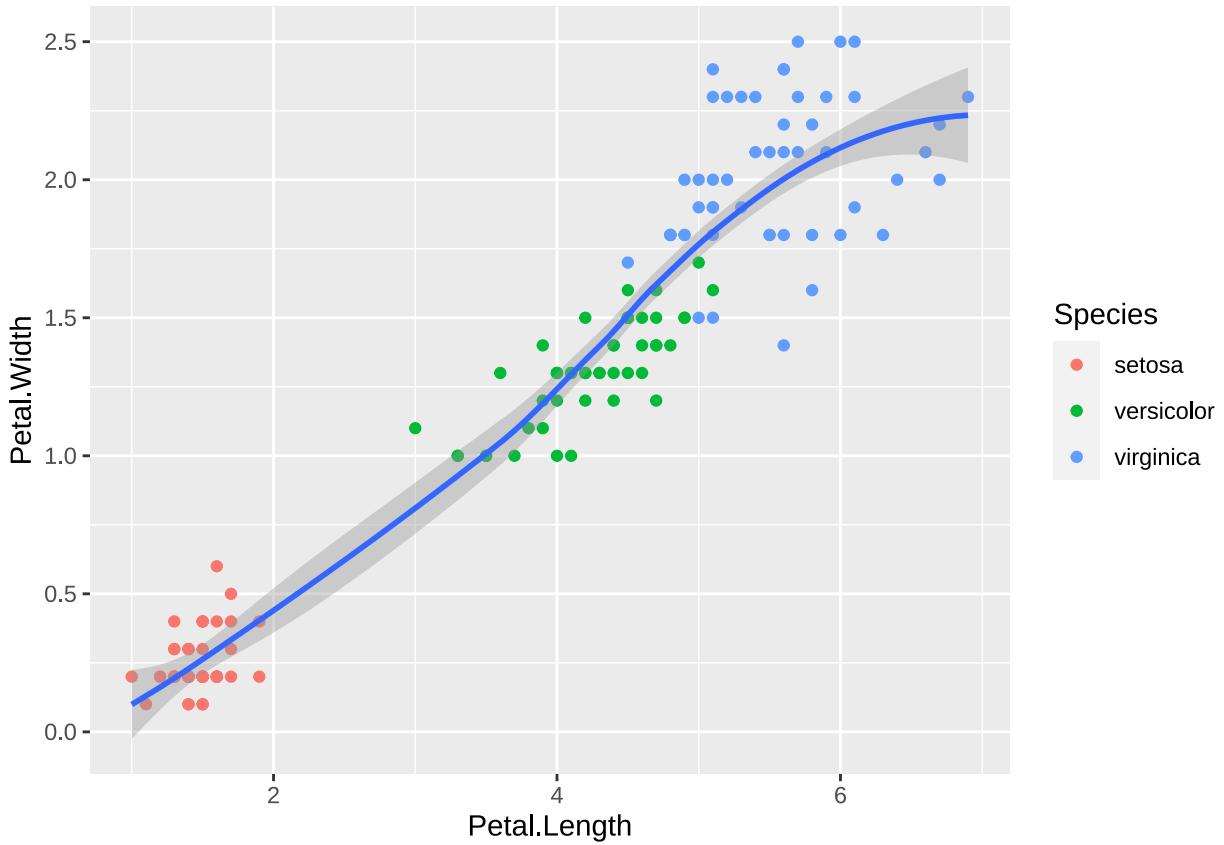
大家可以看到，以上两段代码出来的图是一样。但背后的含义却不同。

例子 (观察两者之间的区别)

```
# 版本一
ggplot(data = iris, mapping = aes(x = Petal.Length, y = Petal.Width, col = Species)) +
  geom_point() + geom_smooth()
```



```
# 版本二
ggplot(data = iris, mapping = aes(x = Petal.Length, y = Petal.Width)) + geom_point(mapping = aes(color = Species))
  geom_smooth()
```



2.4 几何对象

`geom_xxx()` 提供了各种基本图形。列表如下：

- 基础图形：
 - `geom_blank()` 不画图，可以按映射的变量设定坐标范围；
 - `geom_point()` 每个观测为一个散点；
 - `geom_hline()`, `geom_vline()`, `geom_abline()` 画线；
 - `geom_path()` 每个观测提供坐标，在相邻观测之间连线；
 - `geom_ribbon()` 需要 x 和 ymin, ymax 维，在从小到大排序后的相邻观测之间连接阴影区域；
 - `geom_segment()` 需要 x, y 和 xend, yend，为每个观测画一条线段；
 - `geom_rect()` 需要 xmin, xmax, ymin, ymax，为每个观测画一个长方形，可有填充色；
 - `geom_polygon()` 需要 x, y，将相邻观测连续并连接成一个闭合的多边形，中间填充颜色；
 - `geom_text()` 需要 x, y 和 label，每个观测画一条文字标签。
- 单变量图层：
 - `geom_bar()`, `geom_col()` 作条形图；
 - `geom_histogram()` 对连续变量 x 作直方图；

- `geom_density()` 对连续变量 x 作一元密度估计曲线;
 - `geom_dotplot()` 用原点作直方图;
 - `geom_freqpoly()` 用折线作直方图。
- 两变量图形:
 - 两个连续变量 x, y :
 - * `geom_point()` 散点图;
 - * `geom_quantile()` 拟合分位数回归曲线;
 - * `geom_rug()` 在坐标轴处画数值对应的短须线;
 - * `geom_smooth()` 画各种拟合曲线;
 - * `geom_text()` 在指定的 x, y 位置画 label 给出的文字标签;
 - 显示二元分布:
 - * `geom_bin2d()` 作长方形分块的二维直方图;
 - * `geom_density2d()` 作二元密度估计等值线图;
 - * `geom_hex()` 作正六边形分块的二维直方图。
 - 两个变量中有分类变量时:
 - * `geom_count()`: 重叠点越多画点越大;
 - * `geom_jitter()`: 随机扰动散点位置避免重叠, 数值变量有重叠时也可以用;
 - 一个连续变量和一个分类变量:
 - * `geom_col()` 作条形图, 对分类变量的每个值画一个条形, 长度与连续变量值成比例;
 - * `geom_boxplot()` 对每个类做一个盒形图;
 - * `geom_violin()` 对每个类做一个小提琴图。
 - 一个时间变量和一个连续变量:
 - * `geom_area()` 作阴影曲线图, 曲线下方填充阴影色;
 - * `geom_line()` 作折线图, 在相邻两个时间之间连接线段;
 - * `geom_step()` 作阶梯函数图, 在相邻两个时间之间连接阶梯函数线。
 - 不确定性:
 - * `geom_crossbar()` 对每个观测输入的 $x, y, \text{ymin}, \text{ymax}$ 画中间有线的纵向条形;
 - * `geom_errbar()` 对每个观测输入的 $x, \text{ymin}, \text{ymax}$ 画纵向误差条;
 - * `geom_linerange()` 对每个观测输入的 $x, \text{ymin}, \text{ymax}$ 画一条竖线;
 - * `geom_pointrnage()` 对每个观测输入的 $x, y, \text{ymin}, \text{ymax}$ 画一条中间有点的竖线。
 - 地图:
 - `geom_map()`: 用区域边界坐标数据画边界线地图。

- 三个变量：
 - `geom_contour()`: 用输入的 x, y, z 数据画等值线图。
 - `geom_tile()` 用输入的 x, y 位置, width, height 大小和指定的 fill 维画长方形色块填充图。
 - `geom_raster()` 是 `geom_tile()` 的长方形大小相同时的快速版本。

2.4.1 参考书籍

由于这部分内容非常的多，短短两小时不可能讲完，这里给了一些参考资料，各个都是满满的干货。

- 数据科学中的 R-第 14 章 ggplot 之集合对象
- R 语言教程第 30 节-ggplot 各种图形
- Top 50 ggplot2 Visualizations
- Chapter 3: Data Visualisation of *R for Data Science*
- Chapter 28: Graphics for communication of *R for Data Science*
- Graphs in *R Graphics Cookbook*

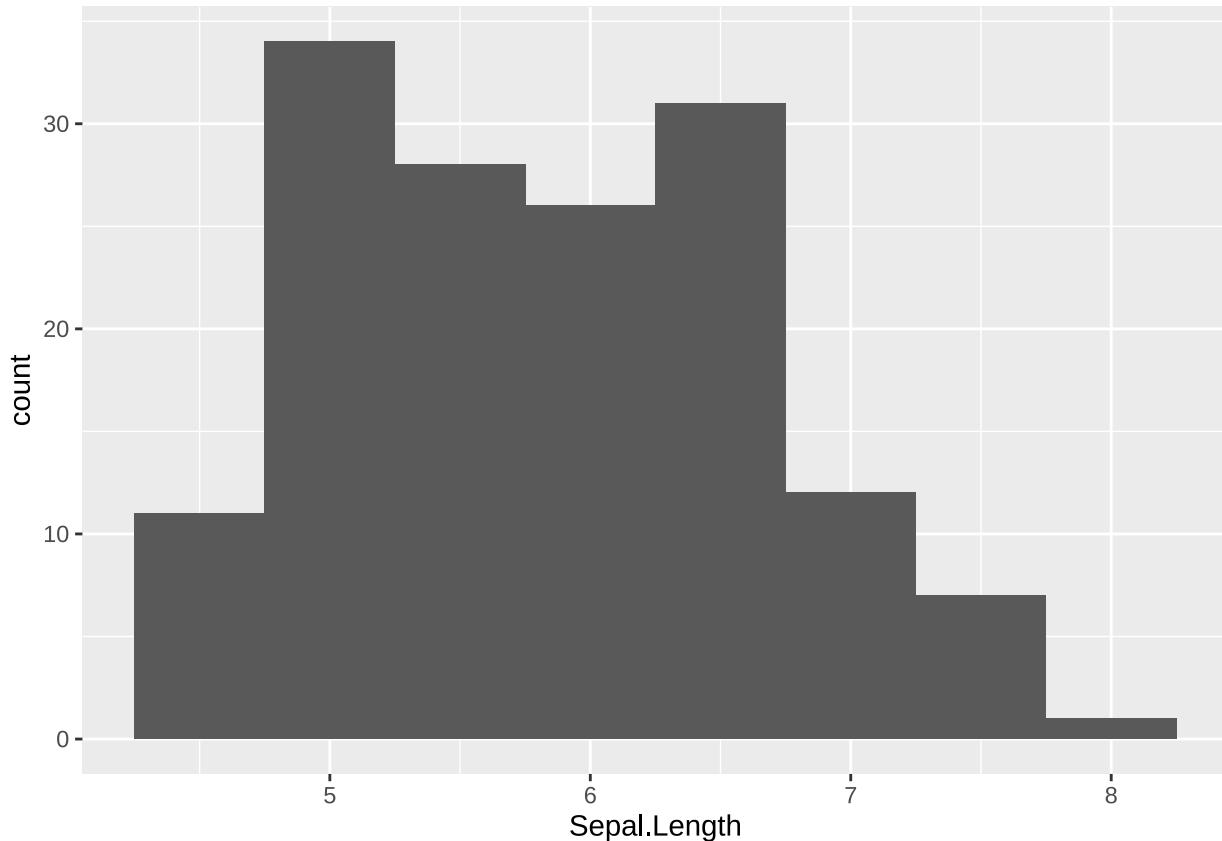
2.5 统计变换

概念：对数据所应用的统计类型/方法。

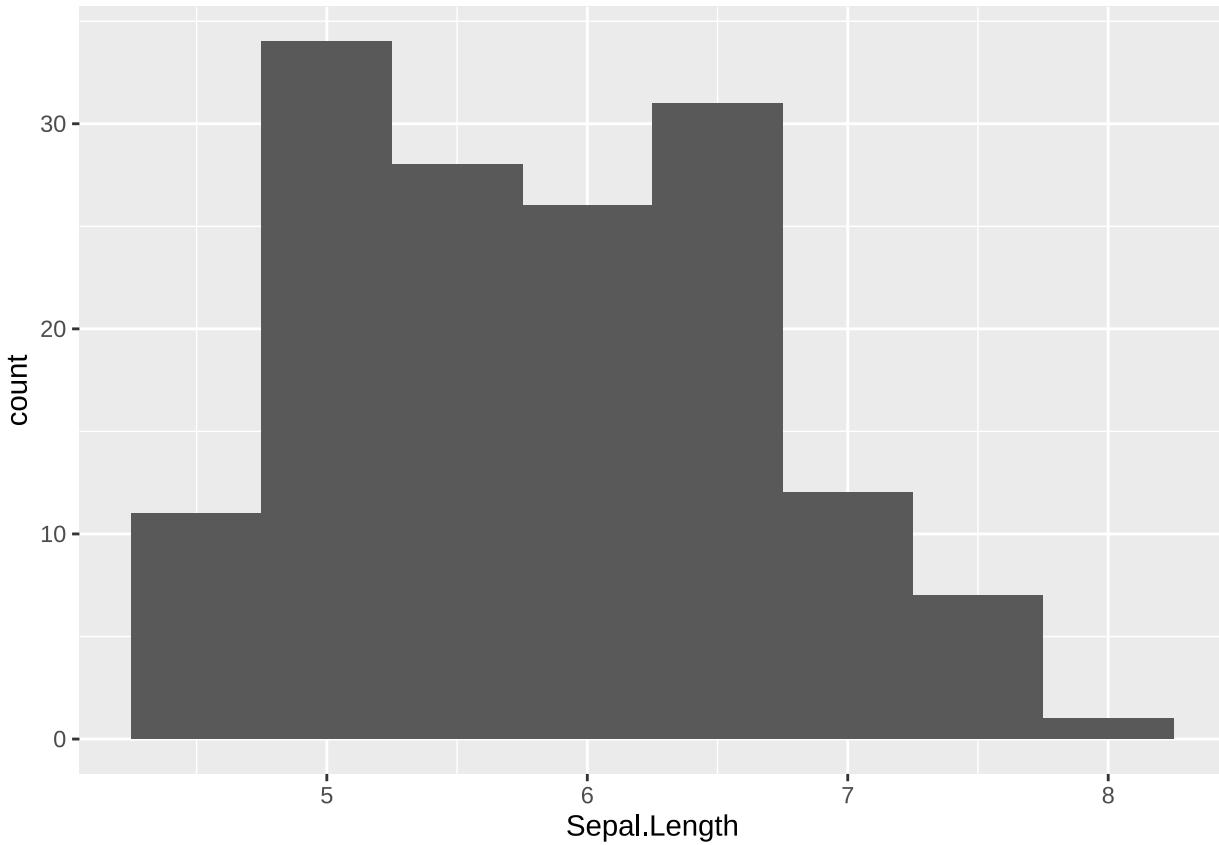
ggplot2 为每一种几何类型指定了一种默认的统计类型，如果仅指定 geom 或 stat 中的一个，另外一个是会自动获取。其中，`stat_identity` 则表示不做任何的统计变换。

示例：只需指定 geom 或 stat 中的一个，具体细小细节可以参考这 <https://bookdown.org/wangminjie/R4DS/ggplot2-stat-layer.html>

```
ggplot(iris) + geom_bar(aes(x = Sepal.Length), stat = "bin", binwidth = 0.5)
```



```
ggplot(iris) + stat_bin(aes(x = Sepal.Length), geom = "bar", binwidth = 0.5)
```



2.6 刻度 scale

这一节我们一起学习 ggplot2 中的 scales 语法，推荐大家阅读 Hadley Wickham 最新版的《ggplot2: Elegant Graphics for Data Analysis》，但如果需要详细了解标度参数体系，还是要看 ggplot2 官方文档

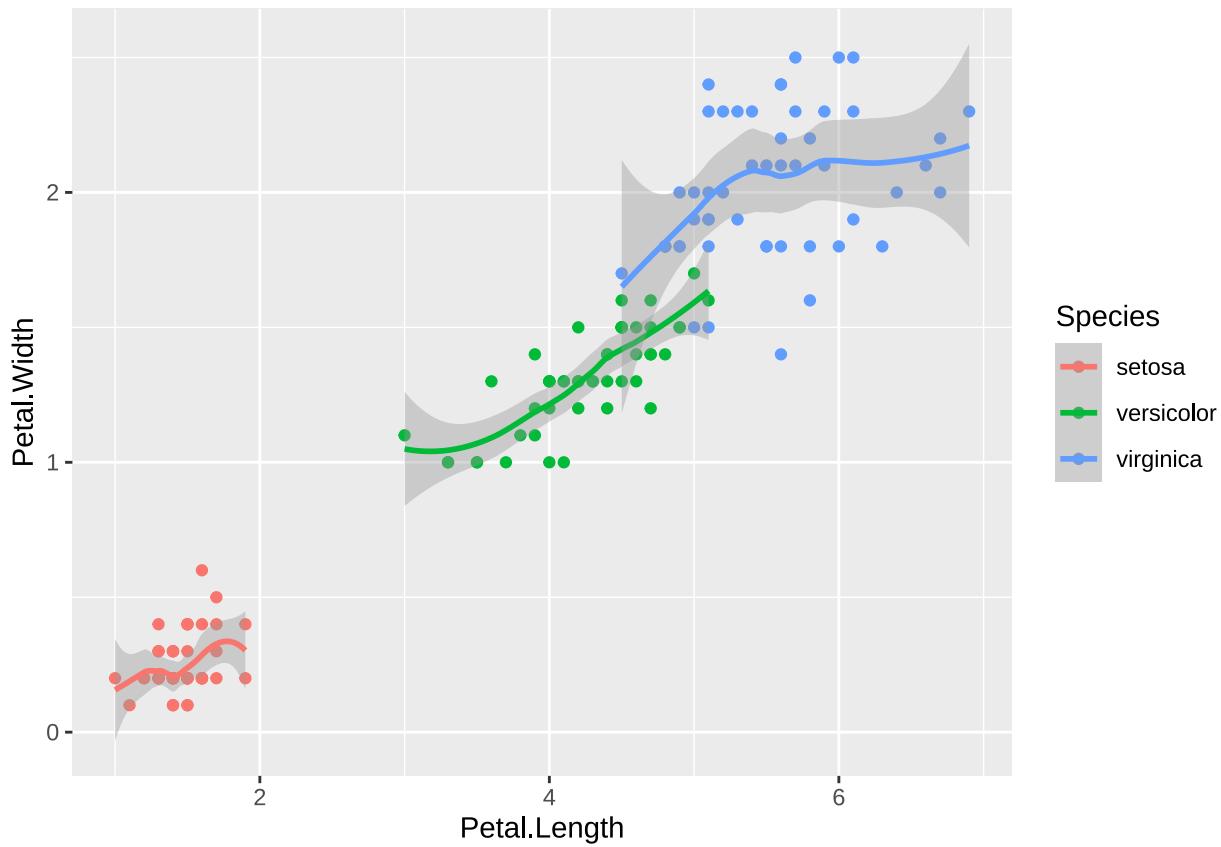
在 `ggplot()` 的 `mapping` 参数中指定 x 维、y 维、color 维等，实际上每一维度都有一个对应的默认刻度 (scale)，即，将数据值映射到图形中的映射方法。

如果需要修改刻度对应的变换或者标度方法，可以调用相应的 `scale_xxx()` 函数。

画图都没用到 scale 啊！

能画个很漂亮的图，那是因为 ggplot2 默认缺省条件下，已经很美观了。（据说 Hadley Wickham 很后悔使用了这么漂亮的缺省值，因为很漂亮了大家都不认真学画图了。马云好像也说后悔创立了阿里巴巴？）

```
# 解释
ggplot(data = iris, mapping = aes(x = Petal.Length, y = Petal.Width, col = Species)) +
  geom_point() + geom_smooth()
```



2.6.1 丰富的刻度体系

注意：标度函数是由“_“分割的三个部分构成的 - scale - 视觉属性名 (e.g., colour, shape or x) - 标度名 (e.g., continuous, discrete, brewer).

- 将数据变量映射到具体的位置、颜色、填充色、大小、符号等。

	连续型	离散型	自定义	同一型
坐标轴标度	scale_x_continuous() scale_x_log10() scale_x_sqrt() scale_x_reverse() scale_x_date() scale_x_datetime() scale_x_time()	scale_x_discrete()		
颜色标度	scale_color_continuous() scale_color_distiller() scale_color_gradient() scale_color_gradient2() scale_color_gradientn() scale_color_viridis_c()	scale_color_discrete() scale_color_brewer() scale_color_hue() scale_color_grey() scale_color_viridis_d()	scale_color_manual()	scale_color_identity()
填充标度	scale_fill_continuous() scale_fill_distiller() scale_fill_gradient() scale_fill_gradient2() scale_fill_gradientn() scale_fill_viridis_c()	scale_fill_discrete() scale_fill_brewer() scale_fill_hue() scale_fill_grey() scale_fill_viridis_d()	scale_fill_manual()	scale_fill_identity()
大小标度	scale_size() scale_size_area() scale_radius()		scale_size_manual()	scale_size_identity()
透明度标度	scale_alpha() scale_alpha_continuous()	scale_alpha_discrete()	scale_alpha_manual()	scale_alpha_identity()
线条标度		scale_linetype() scale_linetype_discrete()	scale_linetype_manual()	scale_linetype_identity()
形状标度		scale_shape()	scale_shape_manual()	scale_shape_identity()
参数列表	name limits breaks labels guide na.value		values	

每个标度函数内部都有丰富的参数系统

```
scale_colour_manual(
  palette = function(),
  limits = NULL,
  name = waiver(),
  labels = waiver(),
  breaks = waiver(),
  minor_breaks = waiver(),
  values = waiver(),
  ...
)
```

- 参数 `name`, 坐标和图例的名字, 如果不想要图例的名字, 就可以 `name = NULL`
- 参数 `limits`, 坐标或图例的范围区间。连续性 `c(n, m)`, 离散型 `c("a", "b", "c")`
- 参数 `breaks`, 控制显示在坐标轴或者图例上的值(元素)
 - 一般情况下, 内置函数会自动完成
 - 也可人工指定一个字符型向量, 与 `breaks` 提供的字符型向量一一对应
 - 也可以是函数, 把 `breaks` 提供的字符型向量当做函数的输入

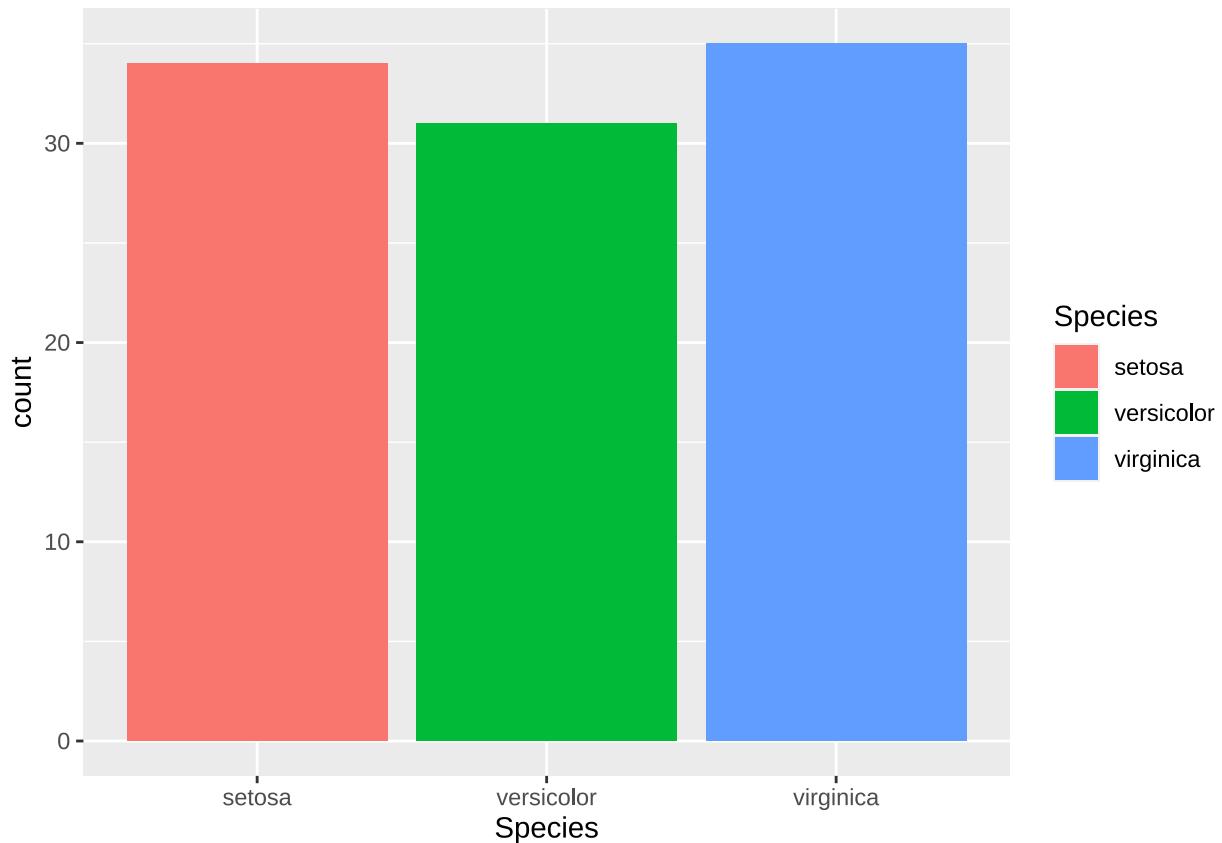
- `NULL`, 就是去掉标签
-
- 参数 `values` 指的是（颜色、形状等）视觉属性值,
 - 要么, 与数值的顺序一致;
 - 要么, 与 `breaks` 提供的字符型向量长度一致
 - 要么, 用命名向量 `c(" 数据标签" = " 视觉属性")` 提供
-
- 参数 `expand`, 控制参数溢出量
-
- 参数 `range`, 设置尺寸大小范围, 比如针对点的相对大小

下面, 我们通过具体的案例讲解如何使用参数, 把图形变成我们想要的模样。

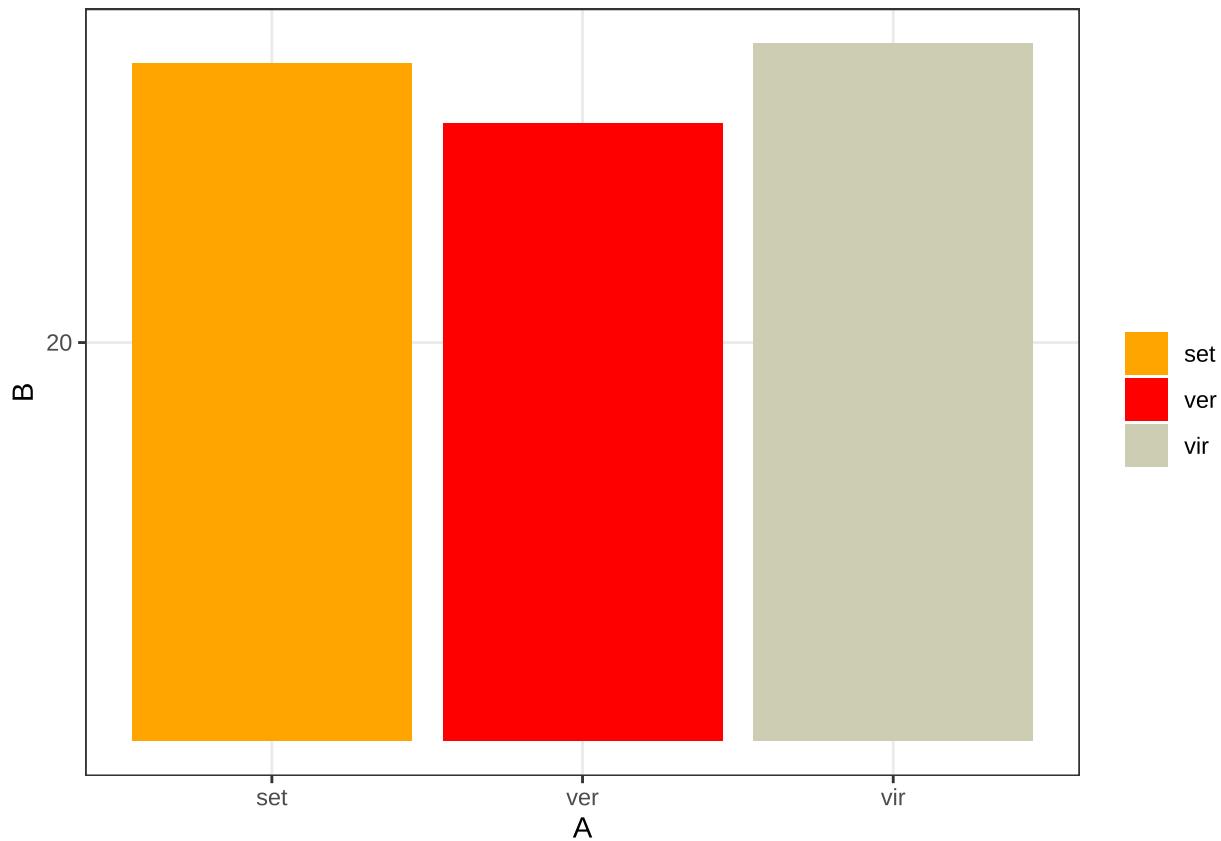
例子: 随机从 `iris` 数据集的 150 个样本中抽取 100 个样本, 并绘制条形图反映 100 个样本中各个鸢尾花种类的数量情况。然后通过修改标尺参数做前后对比图, 进而理解标尺在 `ggplot2` 包中的作用。

```
set.seed(1) # 设置随机种子
my_iris <- iris[sample(1:150, 100, replace = FALSE),] # 随机抽样

p <- ggplot(my_iris) +
  geom_bar(aes(x = Species, fill = Species))
p
```

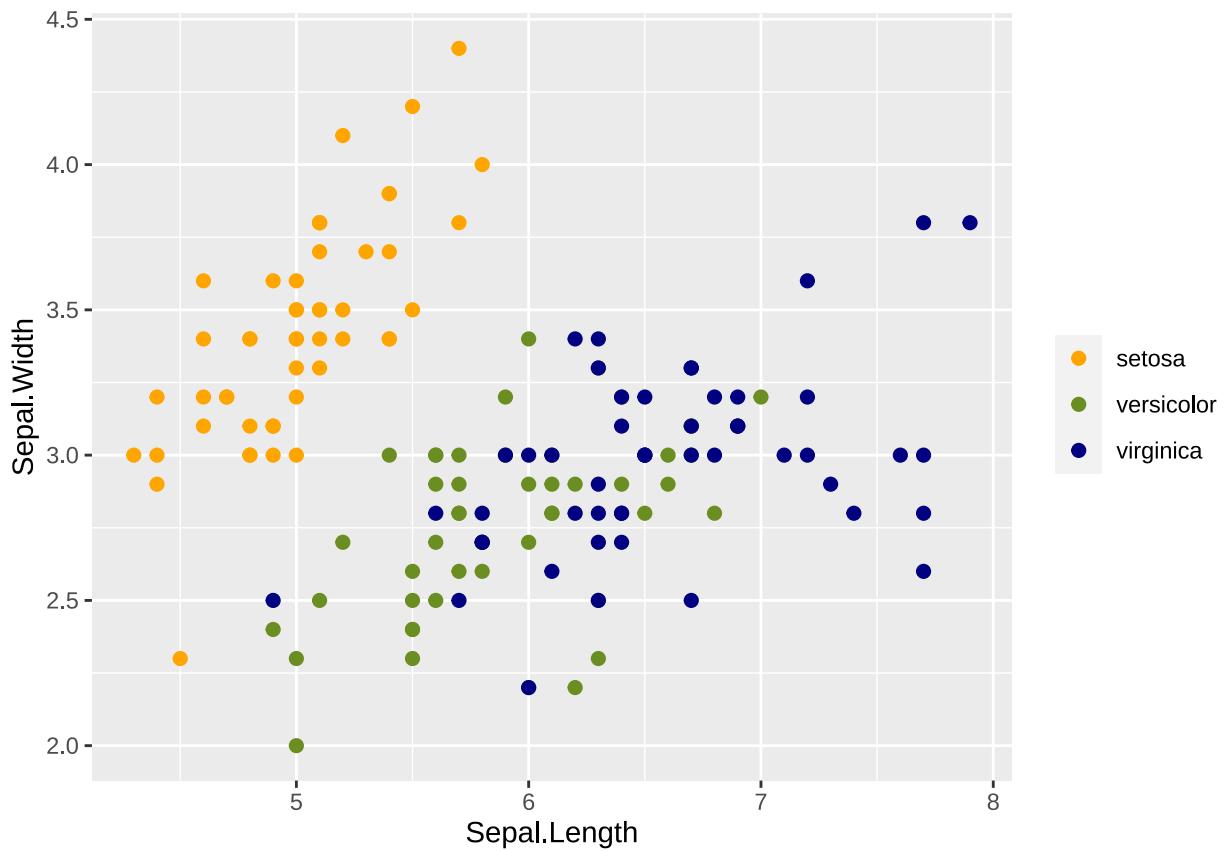


```
p + scale_fill_manual(  
  values = c("orange", "red", "lightyellow3"), # 颜色设置  
  name = NULL, # 图例和轴使用的名称  
  labels = c("set", "ver", "vir") # 图例使用的标签  
) +  
scale_x_discrete(labels = c("set", "ver", "vir"), name = "A") +  
scale_y_continuous(name = "B", breaks = c(20, 40)) +  
theme_bw()
```



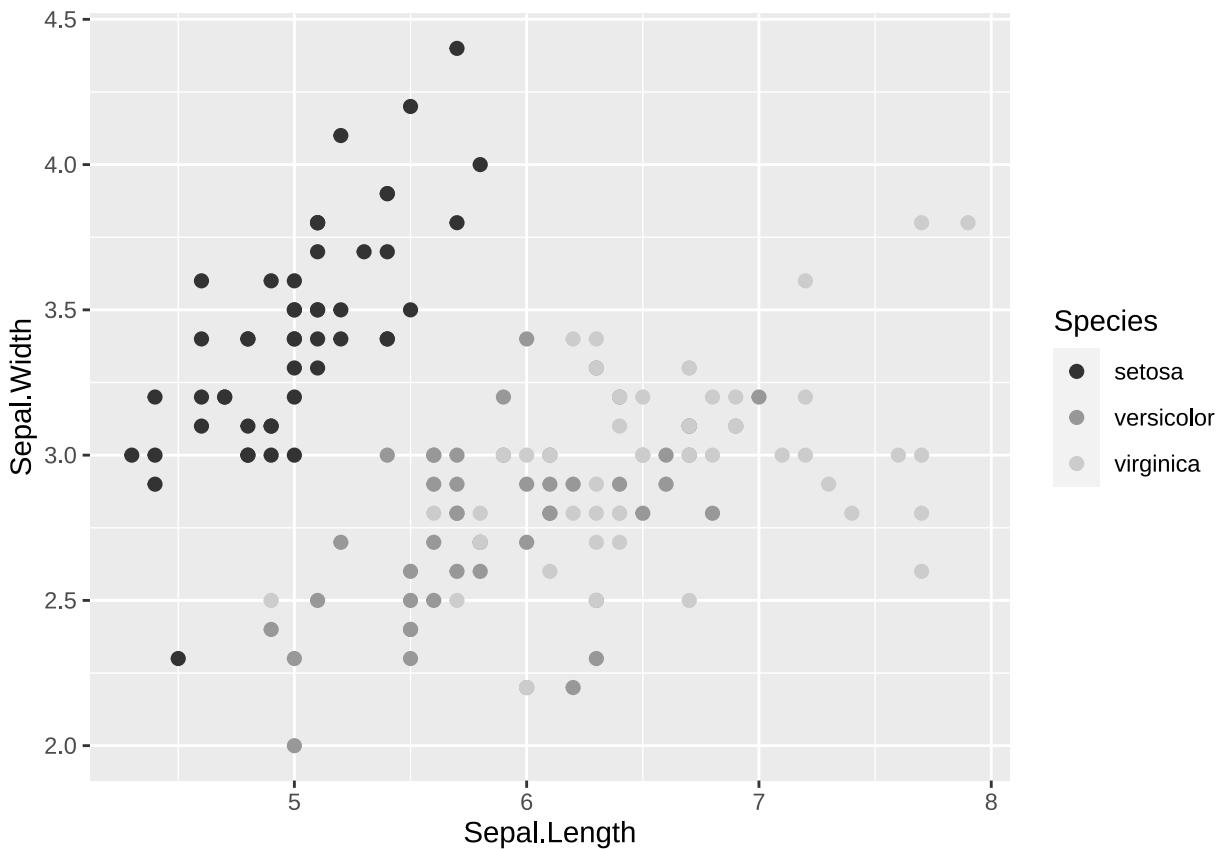
使用 `scale_color_manual` 或 `scale_color_brewer` 函数修改图形的颜色。在对 iris 数据集中的 Sepal.Length 与 Sepal.Width 的散点图分别使用以上两种方法修改散点颜色

```
# 图一: 使用 scale_color_manual 函数
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, colour = Species)) + geom_point(size = 2) +
  scale_color_manual(values = c("orange", "olivedrab", "navy"), name = NULL)
```



图二：使用 `scale_color_brewer` 函数

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, colour = Species)) + scale_color_grey() +  
  geom_point(size = 2)
```



```
# library(RColorBrewer) brewer.pal(3, 'Set1') display.brewer.all()
```

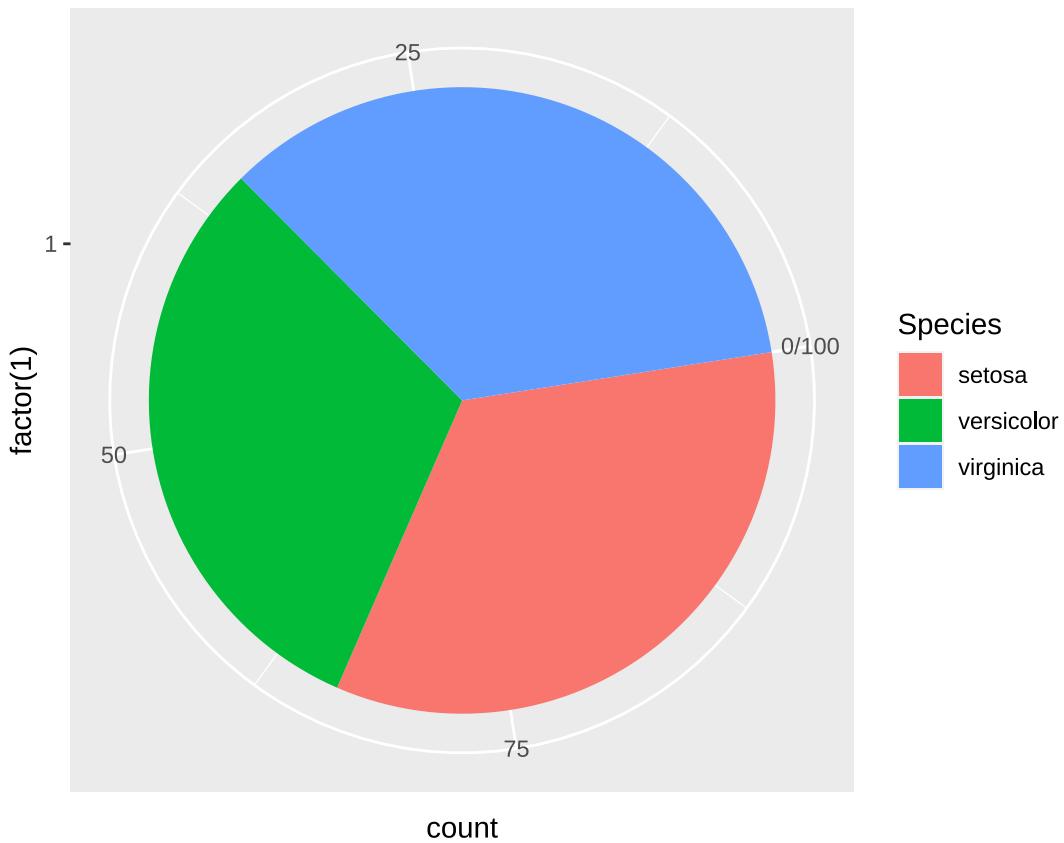
2.7 坐标系

ggplot2 默认的坐标系是笛卡尔坐标系，可以用如下方法指定取值范围: `coord_cartesian(xlim = c(0,5), ylim = c(0, 3))`。

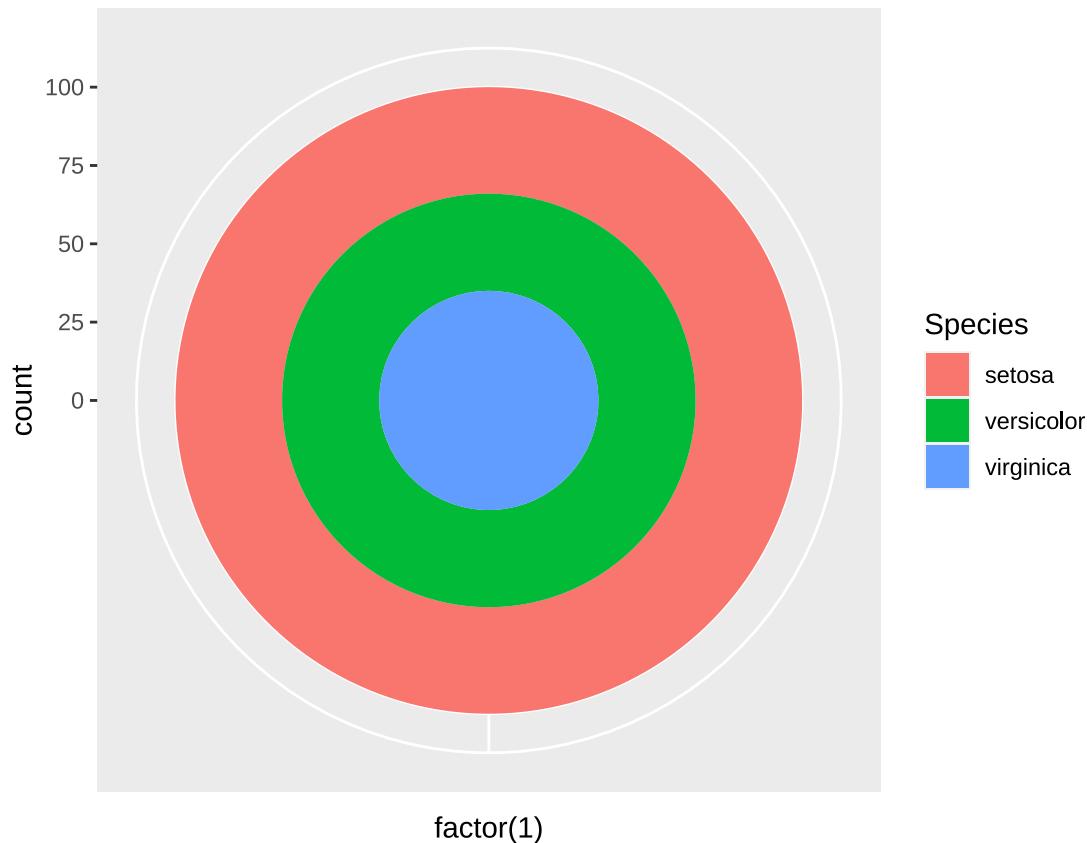
`coord_flip`: x 轴和 y 轴换位置。

`coord_polar(theta = "x", direction=1)` 是角度坐标系, theta 指定角度对应的变量, start 指定起点离 12 点钟方向的偏离值, direction 若为 1 表示顺时针方向, 若为-1 表示逆时针方向。

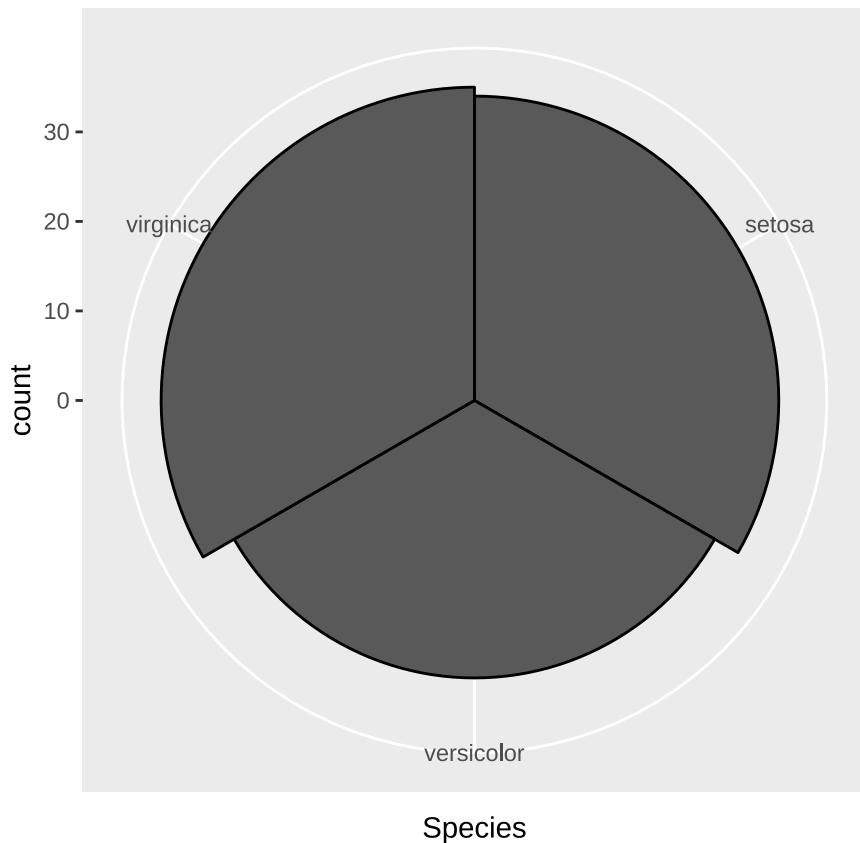
```
# 饼图 = 堆叠长条图 + polar_coordinates
pie <- ggplot(my_iris, aes(x = factor(1), fill = Species)) + geom_bar(width = 1)
pie + coord_polar(theta = "y", direction = -1, start = 30)
```



```
# 靶心图 = 饼图 + polar_coordinates  
pie + coord_polar()
```



```
# 锯齿图 = 柱状图 + polar_coordinates
cxc <- ggplot(my_iris, aes(x = Species)) + geom_bar(width = 1, colour = "black")
cxc + coord_polar()
```



2.8 分面

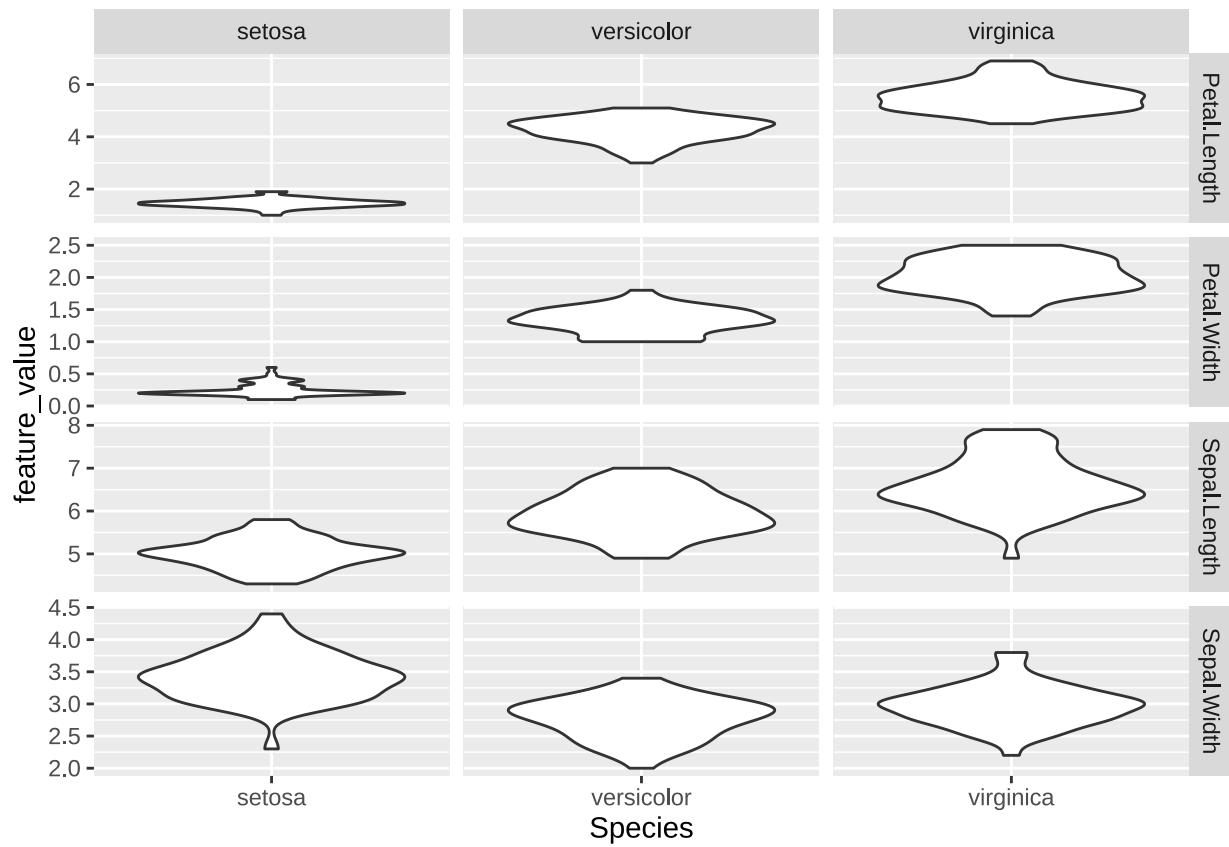
分面，就是分组绘图，根据定义的规则，将数据分为多个子集，每个子集按照统一的规则单独制图，排布在一个页面上。

ggplot2 提供两种分面方法：`facet_grid` 函数和 `facet_wrap` 函数。

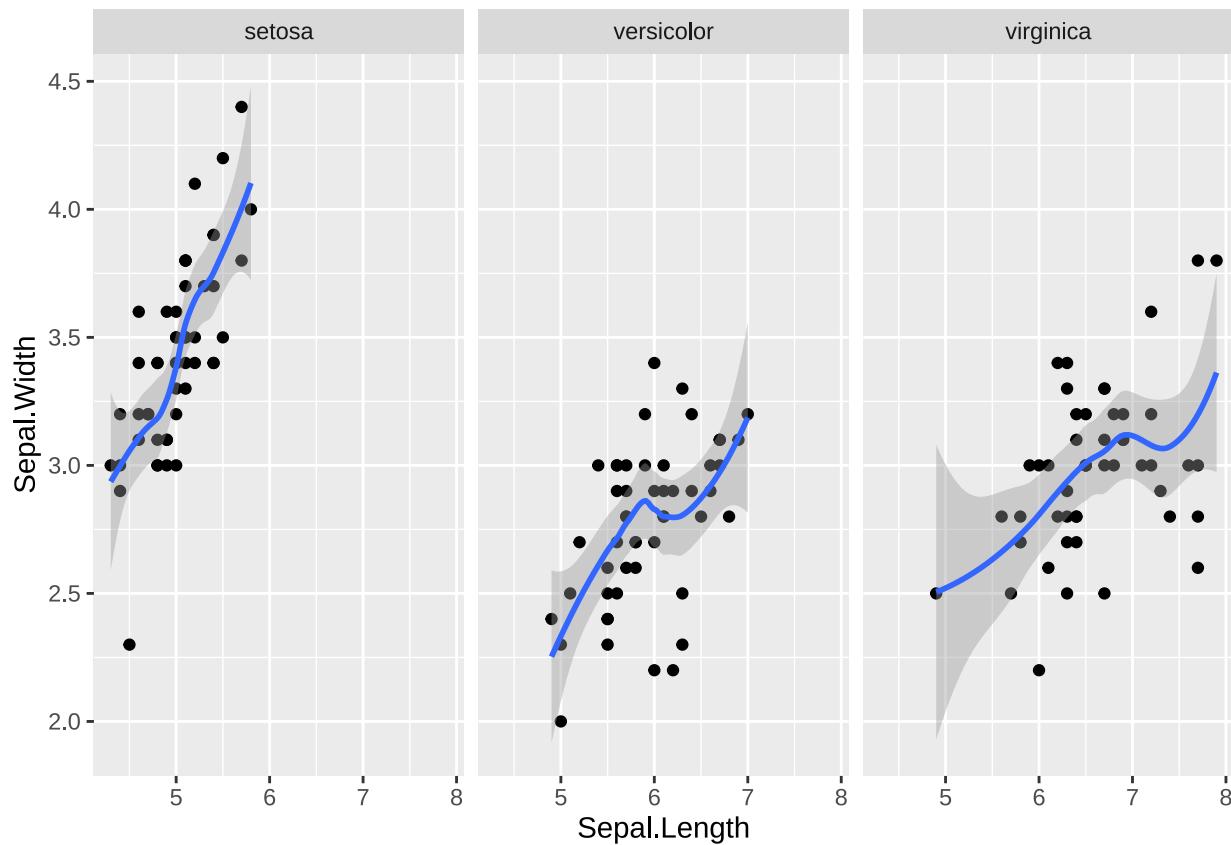
1. `facet_grid` 函数

注意 `facet_grid` 函数是一个二维的矩形布局，每个子集的位置由行位置变量 ~ 列位置变量的决定

```
library(ggplot2)
library(tidyr)
library(dplyr)
my_iris1 <- iris %>% gather(feature_name, feature_value, one_of(c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")))
ggplot(my_iris1) +
  geom_violin(aes(x = Species, y = feature_value)) + # 绘制小提琴图
  facet_grid(feature_name ~ Species, scales = "free") # 分面
```



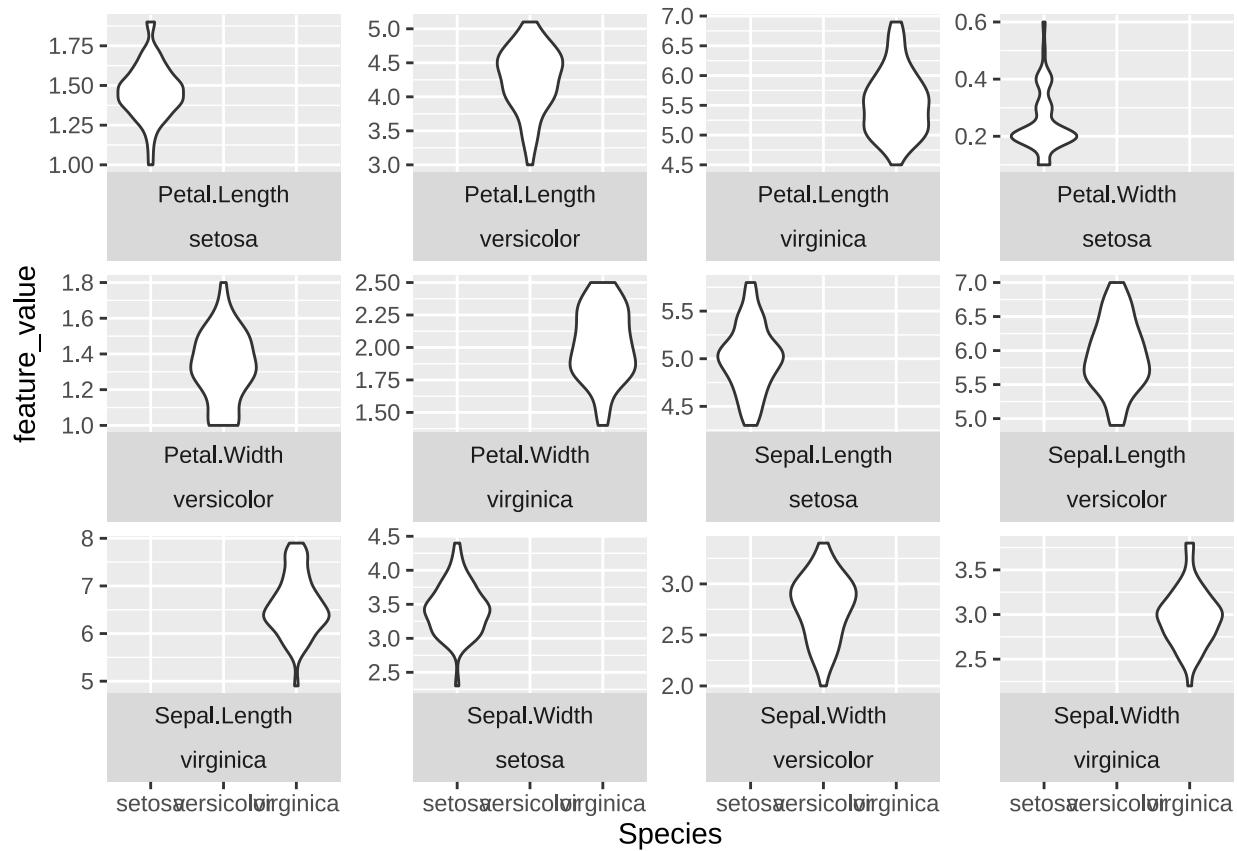
```
# iris 例子
ggplot(data = iris, mapping = aes(x = Sepal.Length, y = Sepal.Width)) + # 底层画布
  geom_point() +
  geom_smooth() +
  facet_grid(~Species)
```



2. facet_wrap 函数

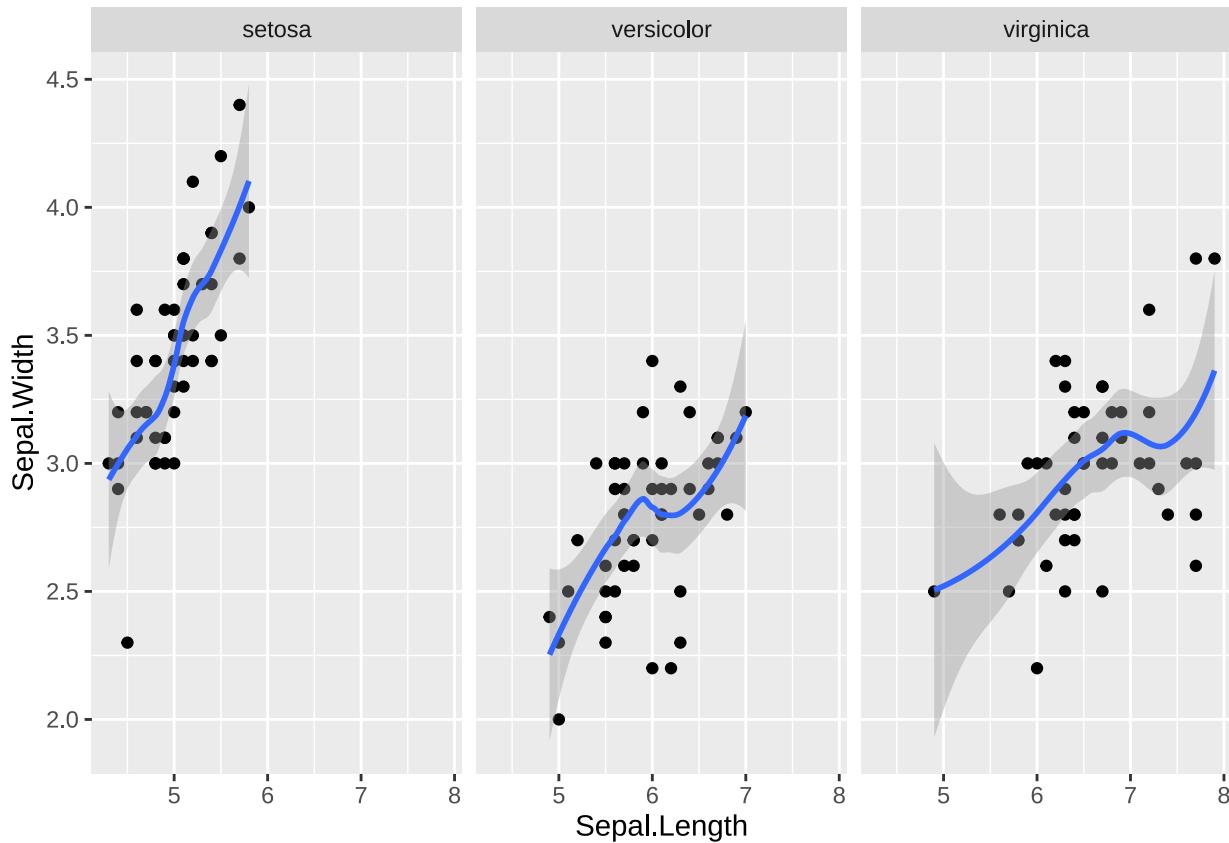
facet_wrap 函数生成一个动态调整的一维布局，根据“~ 位置变量 1+ 位置变量 2+...”来确定每个子集的位置，先逐行排列，放不下了移动到下一行。

```
ggplot(my_iris1) + geom_violin(aes(x = Species, y = feature_value)) + facet_wrap(~feature_name +
  Species, scales = "free_y", nrow = 3, strip.position = "bottom")
```



```
# iris 例子
```

```
ggplot(data = iris, mapping = aes(x = Sepal.Length, y = Sepal.Width)) + # 底层画布
  geom_point() +
  geom_smooth()+
  facet_wrap(~Species)
```



2.9 标题、标注、指南、拼接

除了 `ggplot()` 指定数据与映射, `geom_xxx()` 作图, 还可以用许多辅助函数增强图形。

- `labs()` 可以设置适当的标题和标签。
- `annotate()` 函数可以直接在坐标系内进行文字、符号、线段、箭头、长方形的绘制。
- `guides()` 函数可以控制图例的取舍以及做法。
- `theme()` 函数可以控制一些整体的选项如背景色、字体类型、图例的摆放位置等。

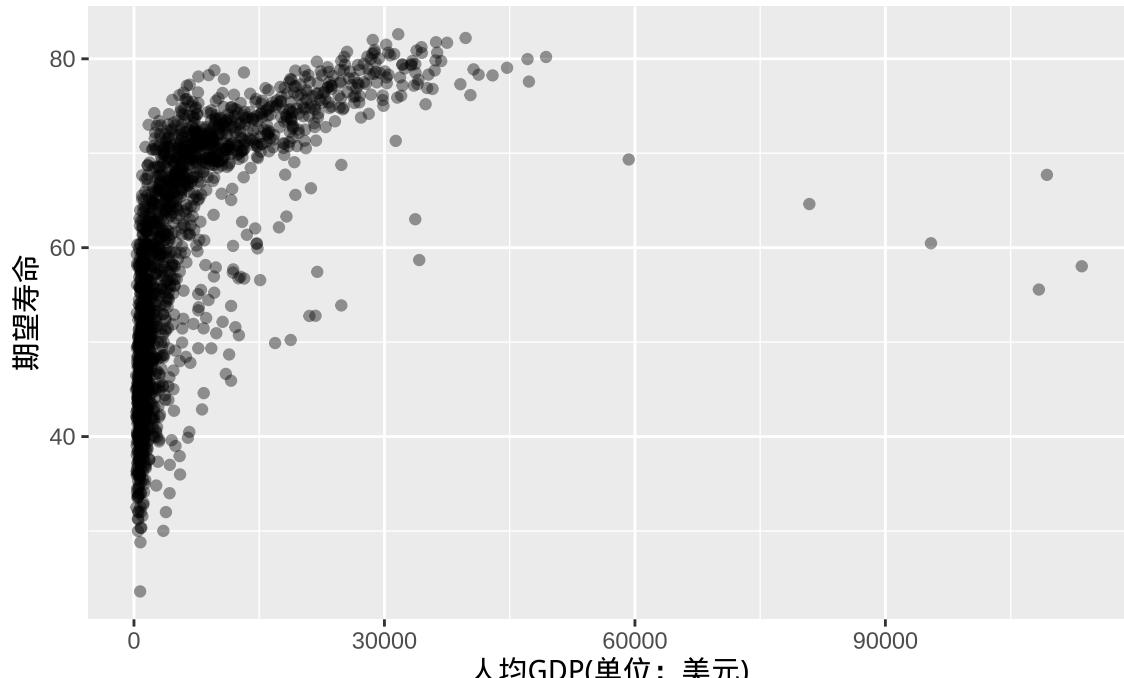
2.9.1 标题

函数 `labs()` 可以用来指定图形上方的标题 (`title`)、副标题 (`subtitle`)、右下方的标注 (`caption`)、左上方的标签以及坐标轴标题和其它维的名称。例如:

```
library(gapminder)
p <- ggplot(data = gapminder, mapping = aes(x = gdpPercap, y = lifeExp))
p + geom_point(alpha = 0.4) + labs(title = " 各国各年度人均 GDP 与期望寿命的关系",
  subtitle = "1952-2007", tag = " 散点图", caption = " 数据来源: gapminder",
  x = " 人均 GDP(单位: 美元)", y = " 期望寿命")
```

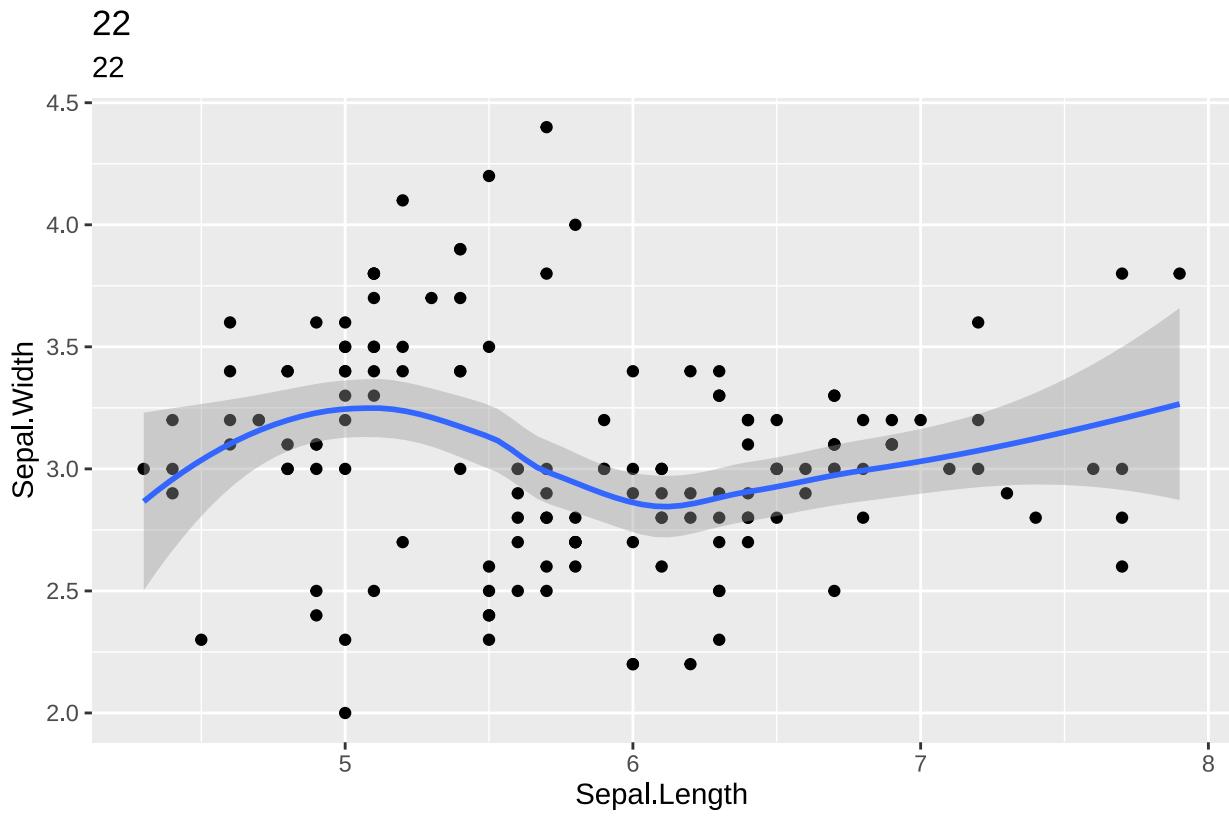
散点图

各国各年度人均GDP与期望寿命的关系
1952-2007



#iris 案例

```
ggplot(data = iris, mapping = aes(x = Sepal.Length, y = Sepal.Width)) + # 底层画布
  geom_point() +
  geom_smooth() +
  labs(
    title = "22",
    subtitle = "22",
    caption = "22"
  )
```



22

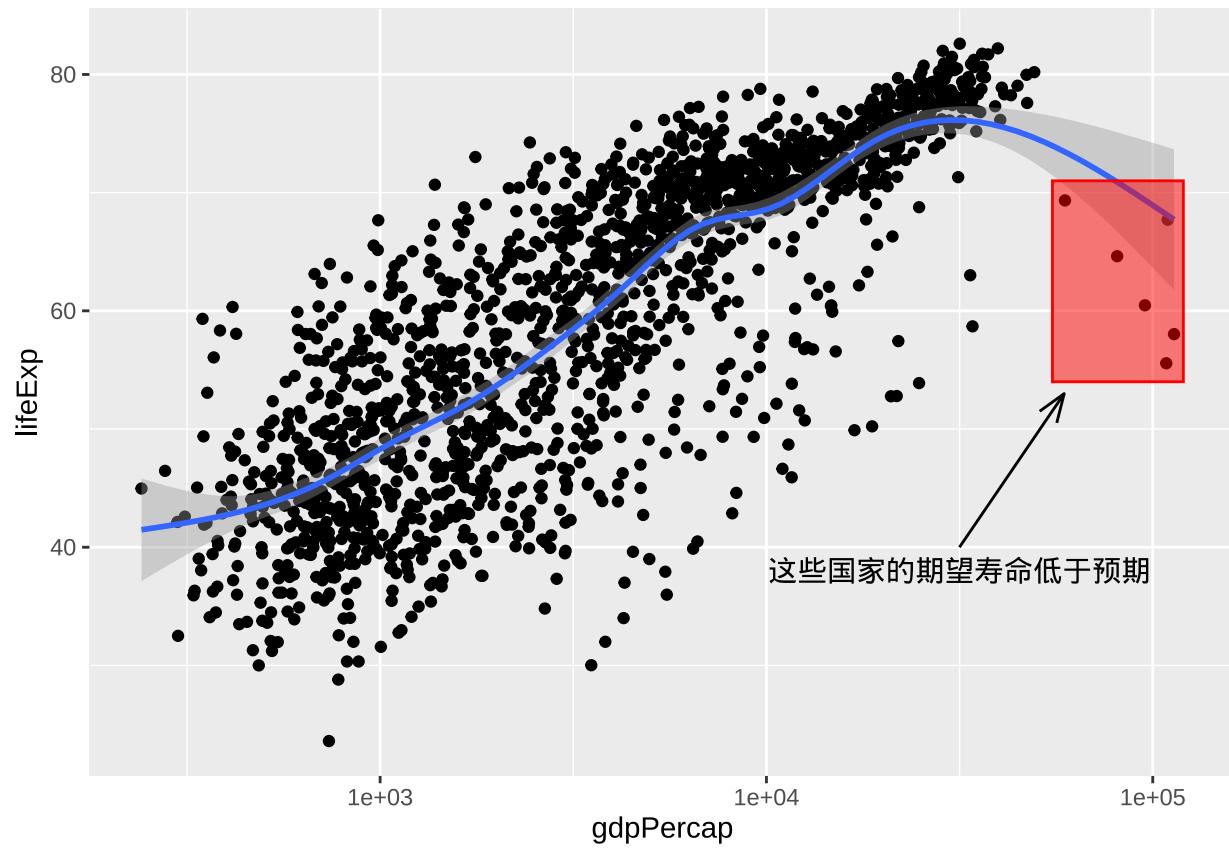
`labs()` 只是提供了这些标题功能，一般并不会同时使用这些功能。在出版图书内，图形下方一般伴随有图形说明，这时一般就不再使用标题、副标题、标签、标注，而只需写在图的伴随说明文字中，当然，坐标轴标签一般还是需要的。

2.9.2 标注功能

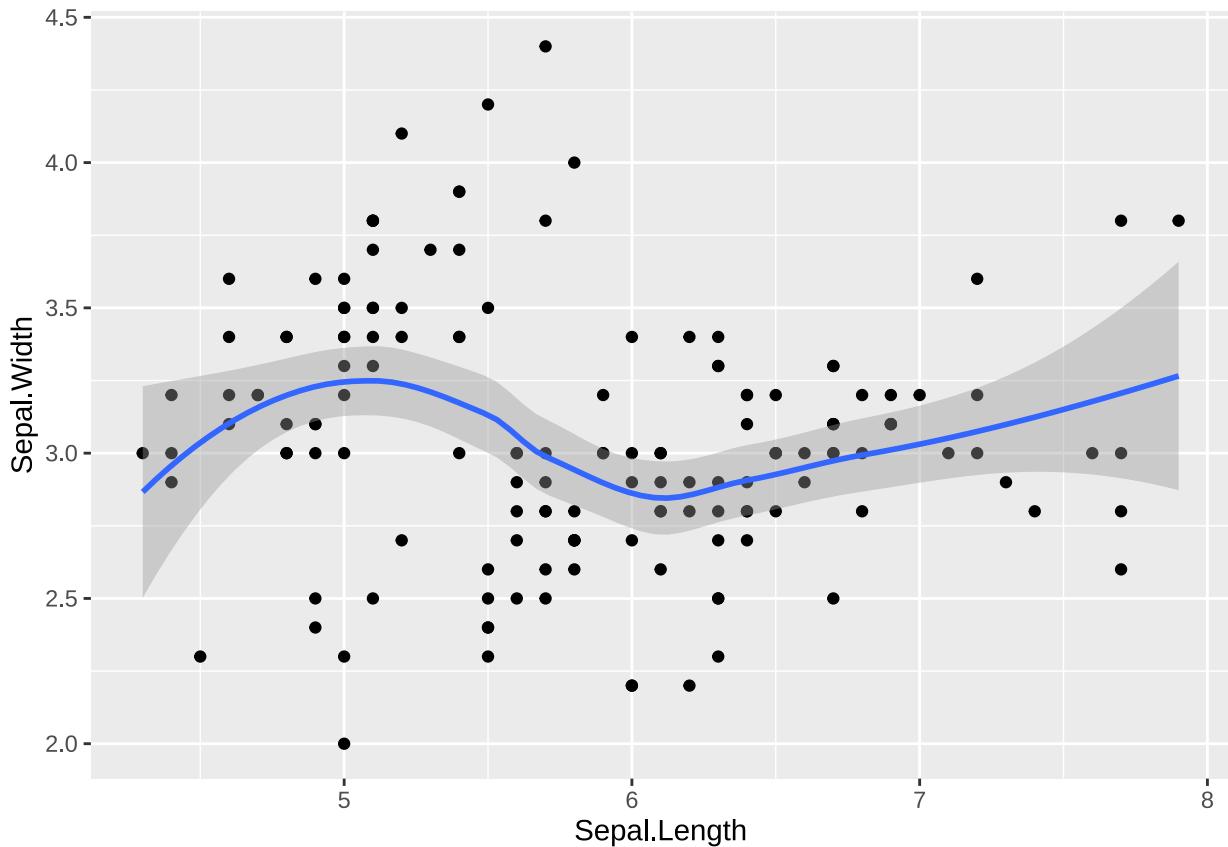
通过 `annotate(geom = "text")` 调用 `geom_text()` 的功能，可以在一个散点图中标注多行文字，多行之间用"`\n`" 分开：

在 `annotate()` 中选 `geom="rect"`，给出长方形的左右和上限界限，可以将上面图形中最右侧偏低的点用长方形填充标出。可以在 `annotate()` 中选 `geom="line"` 画线，需要给出线的起点和终点坐标，可以 `arrow` 选项要求画箭头，用 `arrow()` 函数给出箭头的大小、角度等设置，如：

```
p + geom_point() + geom_smooth(method = "gam") + scale_x_log10() + annotate(geom = "rect",
  xmin = 55000, xmax = 120000, ymin = 54, ymax = 71, col = "red", fill = "red",
  alpha = 0.5) + annotate(geom = "line", x = c(59000, 31600), y = c(53, 40), arrow = arrow(angle =
  length = unit(4, "mm")))) + annotate(geom = "text", x = 31600, y = 38, label = "这些国家的期望寿命")
```



```
#iris 例子 + annotate, hline, abline
ggplot(data = iris, mapping = aes(x = Sepal.Length, y = Sepal.Width)) + # 底层画布
  geom_point() +
  geom_smooth()
```



可以用 `geom_hline()`、`geom_vline()` 和 `geom_abline()` 画横线、竖线、斜线。ggplot2 的默认主题会自动画参考线，可以用 `theme()` 函数指定参考线画法。

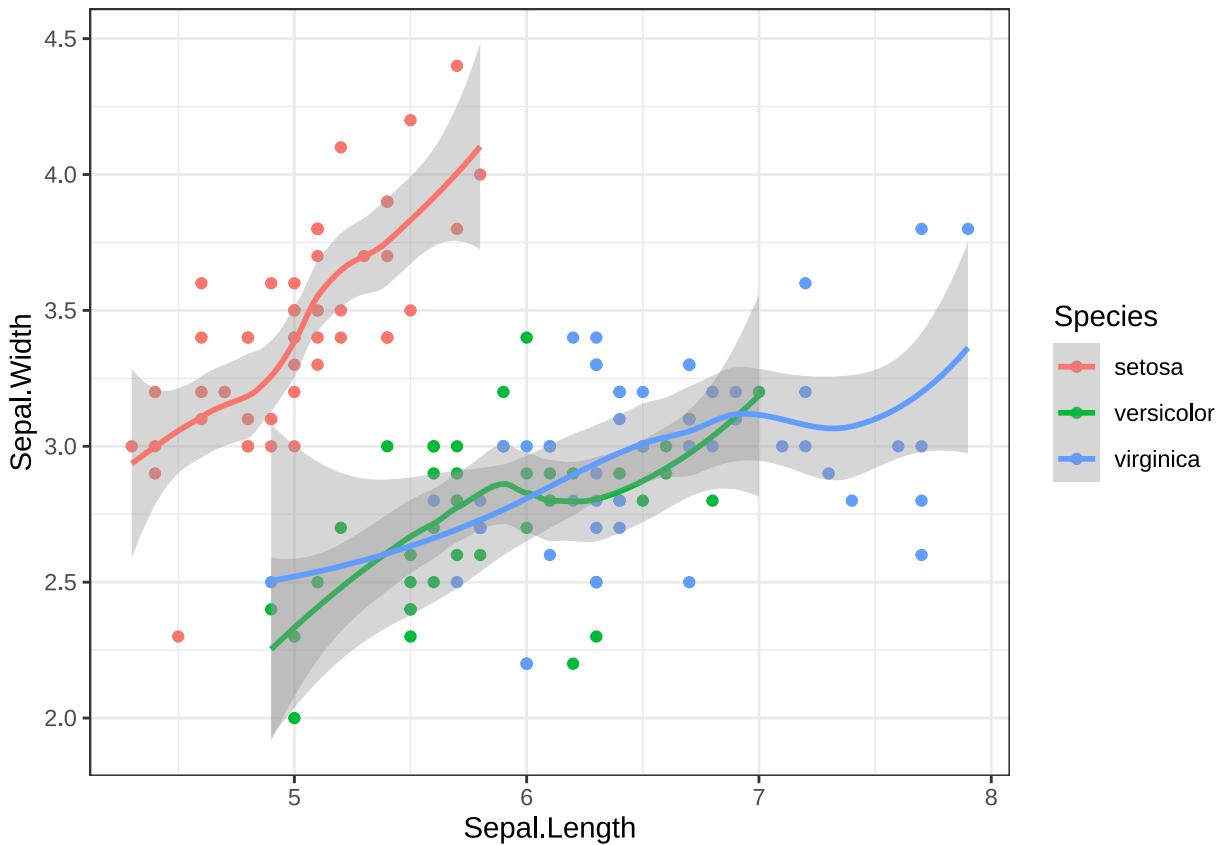
2.9.3 指南

对于颜色、填充色等维度，会自动生成图例。用 `guides(color = FALSE)` 这样的方法可以取消指定维度的图例。

`theme()` 可以调整一些整体的设置，如背景色、字体、图例的摆放位置。

例如：用 `theme()` 的 `legend.position` 改变图例的位置，如 `theme(legend.position = "top")` 可以将图例放置在上方，默认是放置在右侧的。可取值有"none"、"left"、"right"、"bottom"、"top"，如：

```
#iris 例子
ggplot(data = iris, mapping = aes(x = Sepal.Length, y = Sepal.Width, col = Species)) + # 底层画布
  geom_point() +
  geom_smooth() +
  theme(legend.position = 'left', panel.background = element_blank()) + theme_bw()
```



2.9.4 主题

ggplot2 包作图可以实现内容与设计的分离，这里内容就是指数据、映射、统计、图形类型等方面，而设计就是指背景色、颜色表、字体、坐标轴做法、图例位置等的安排。将作图任务分解为内容与设计两个方面，可以让数据科学家不必关心设计有关的元素，而设计可以让专门的艺术设计人才来处理。这种工作分配已经在图书出版、网站、游戏开发等行业发挥了重要作用。

`theme()` 函数用来指定设计元素，称为主题 (theme)，而且可以单独开发 R 扩展包来提供适当的主题。**ggthemes** 扩展包是一个这样的包。

`theme_set()` 可以改变后续 ggplot2 作图的主题(配色、字体等)。如 `theme_set(theme_bw())`, `theme_set(theme_dark())` 等。对单次绘图，可以直接用加号连接 `theme_gray()` 等这些主题函数。主题包括 `theme_gray()` (默认主题)、`theme_minimal()`、`theme_classic()` 等。

`theme()` 函数还可以直接指定颜色、字体、大小等设置。

```
# iris 例子
```

2.10 保存图片

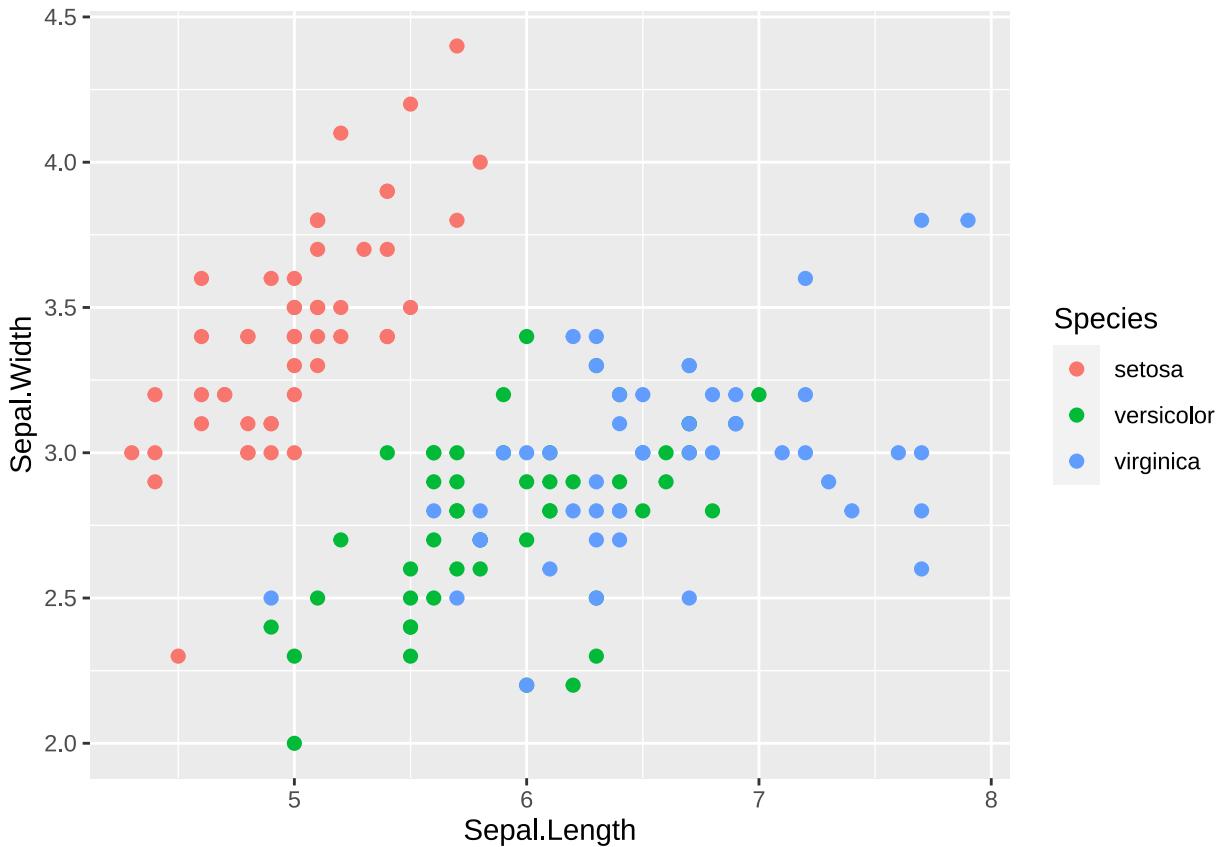
ggplot2 包中提供 ggsave 函数进行图形保存。ggsave 函数的使用格式如下所示。

```
ggsave(filename, width, height, ...)
```

其中, filename 为保存的文件名与路径, width 指图像宽度, height 指图像高度。

示例: 运行下列代码将会在当前工作目录下生成一个名为 mygraph 的 pdf 图形。

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, colour = Species)) + geom_point(size = 2)
```



```
ggsave(file = "mygraph1.png", width = 6, height = 8)
```

或者可以使用 Rstudio 界面进行保存图片, 具体教程课件 (R 语言可视化基础教程)

2.11 例子

该部分来源于: 公众号 [小明的数据分析笔记本]。大家可以通过以下例子对今天所学的知识进行回顾。

```

## 小明推送笔记《小明的数据分析笔记本》
# 跟着 Nature microbiology 学画图 ~R 语言 ggplot2 画柱形图
# https://mp.weixin.qq.com/s/E-1X_VSq03AhvC_0cNEgyQ

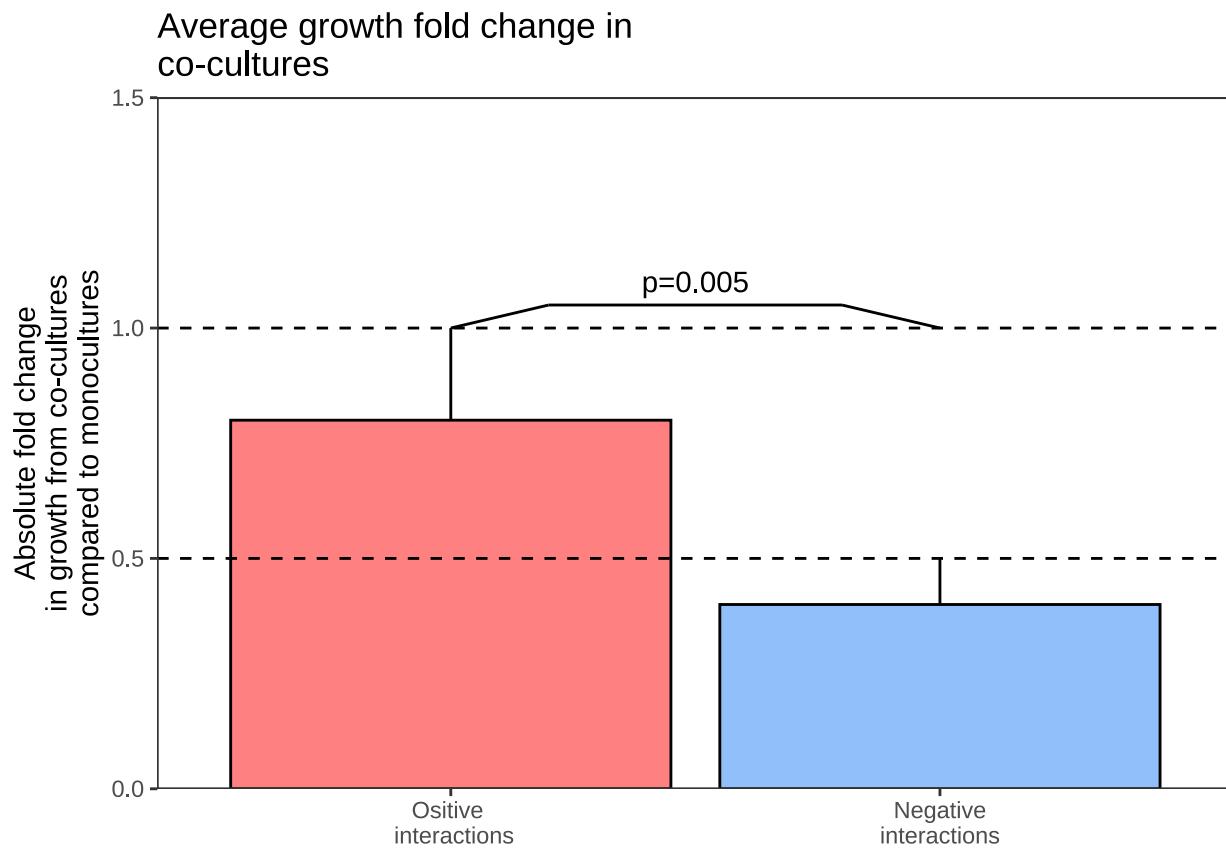
library(ggplot2)

#### 柱状图 + 误差项
data = data.frame("group" = c("A", "B"), "value" = c(0.8, 0.4), "errorbar" = c(.2, .1))
data

##   group value errorbar
## 1      A     0.8      0.2
## 2      B     0.4      0.1

p1 = ggplot(data, aes(x = group, y = value)) +
  geom_col(aes(fill=group), color="black") + # 柱状图
  geom_hline(yintercept = 1, lty = 2) + # 加横线
  geom_hline(yintercept = 0.5, lty = "dashed") +
  theme_bw() + # 主题设置
  theme(panel.grid = element_blank(), # 网格为空
        legend.position = "none") + #legend 位置为无，就是不加
  scale_y_continuous(expand = c(0,0), limits = c(0,1.5)) + #y 为连续，设置 ylim
  scale_x_discrete(label = c("Positive \n interactions", "Negative\n interactions")) + #x 为离散
  annotate("segment", x=1, y=0.8, xend=1, yend=1) + # 加线段 segment，当然这个函数可以加很多其他的包括字
  annotate("segment", x=2, y=0.4, xend=2, yend=0.5) +
  labs(x = NULL, # 标签，注意 \n 可以空行
       y = 'Absolute fold change\nin growth from co-cultures\ncompared to monocultures',
       title = "Average growth fold change in\nco-cultures") +
  annotate("segment", x=1, y=1, xend=1.2, yend=1.05) +
  annotate("segment", x=2, y=1, xend=1.8, yend=1.05) +
  annotate("segment", x=1.2, y=1.05, xend=1.8, yend=1.05) +
  annotate("text", x = 1.5, y = 1.1, label = "p=0.005") +
  scale_fill_manual(values = c("#ff8080", "#90bff9")) # 填充色使用离散颜色 manual，两种颜色这里。
p1

```



2.11.2 有正值和负值的柱形图

```
## 有正值和负值的柱形图

x <- 1:28
y <- sample(-100:150, 28, replace = F)
df2 <- data.frame(x, y)
df2$x = as.factor(df2$x)
df2$group <- ifelse(df2$y > 0, "A", "B")
df2$group<-factor(df2$group,
                    labels = c("Synergistic interactions",
                               "Non-synergistic interactions"))

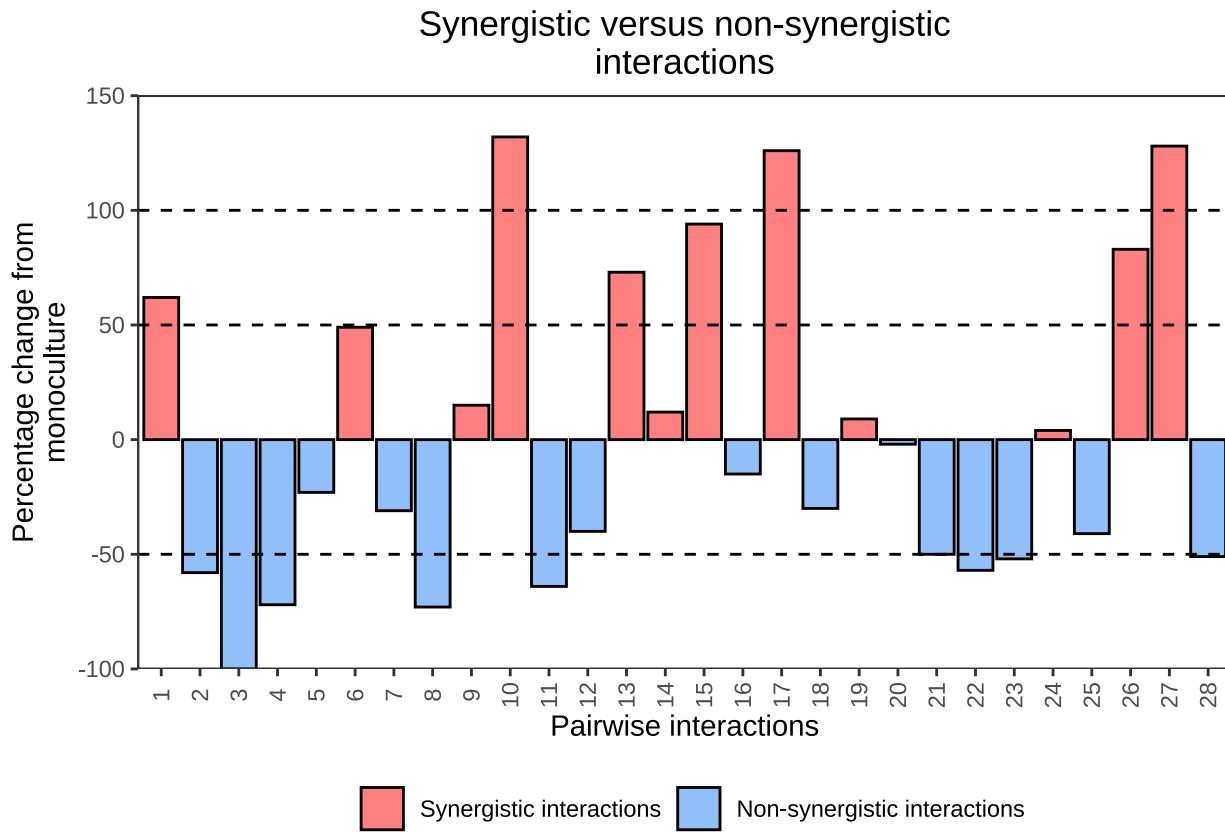
head(df2)
```

```
##   x     y             group
## 1 1    62  Synergistic interactions
## 2 2   -58 Non-synergistic interactions
## 3 3  -100 Non-synergistic interactions
```

```
## 4 4 -72 Non-synergistic interactions
## 5 5 -23 Non-synergistic interactions
## 6 6 49      Synergistic interactions

p2 = ggplot(df2,aes(x,y)) +
  geom_col(aes(fill = group),col = "black") +
  geom_hline(yintercept = 100,lty = "dashed") +
  geom_hline(yintercept = 50,lty = "dashed") +
  geom_hline(yintercept = -50,lty = "dashed") +
  theme_bw() +
  theme(panel.grid = element_blank(),
        axis.text.x = element_text(angle = 90,hjust = 0.5,
                                    vjust = 0.5),
        plot.title = element_text(hjust = 0.5),
        legend.position = "bottom",
        legend.title = element_blank()) + # 取消标签的名称
  scale_y_continuous(expand = c(0,0),
                     limits = c(-100,150),
                     breaks = c(-100,-50,0,50,100,150)) +
  labs(x="Pairwise interactions",
       y="Percentage change from\nmonoculture",
       title = "Synergistic versus non-synergistic\ninteractions") + # 标签说明
  scale_fill_manual(values = c("#ff8080","#90bff9"))

p2
```



2.11.3 合并两图

合并两图或者多图可以使用以下包：

- cowplot 包的 plot_grid()
- pathwork 包
- gridExtra 包的 grid.arrange()

具体可以参考我公众号的这篇推文R 可视乎 | 合并多幅图形

这里使用了 cowplot 包

```
## 合并两图 (使用 cowplot 包)
library(cowplot)
pdf("test/plot_cow.pdf", width = 8, height = 4)
plot_grid(p1, p2, ncol = 2, nrow = 1, labels = c("d", "e"))
dev.off()

## pdf
## 2
```

Chapter 3

其他相关拓展包

介绍一些 ggplot2 扩展可视化包以及其他实用的包。全部在这：其他相关拓展包

3.1 ggplot 官网 108 种拓展包

官网一共汇总了 108 种拓展的 ggplot 包ggplot81 种拓展包

3.2 我整理的 11 个扩展包

1. ggviz 包—数据可视化交互
2. ggridges 包—峰峦图详细介绍
3. esquisse 包—不写代码生成 ggplot 图
4. calendR 包—私人定制专属日历
5. corrplot 包：相关性矩阵可视化
6. cowplot 包：用 R 添加水印
7. flexdashboard 包：用于 R 的简单交互式仪表盘
8. gghalves 包-你五毛我五毛
9. 用 ggpibr 包制图
10. reticulate 包——数据科学者的福音
11. igraph 包——绘制网络图
12. ggthemes待补充

13. `ganimate`待补充

Chapter 4

认识交互式绘图工具

前面可视化的结果就是一个静态的图形，所有信息都一目了然地放在一张图上。

静态图形适合于分析报告等纸质媒介，而在网络时代，如果在网页上发布可视化，那么动态的、交互的图形则更有优势。

在 R 的环境中，动态交互图形的优势在于能和 knitr，shiny 等框架整合在一起，能迅速建立一套可视化原型系统。

由于 pdf 不支持 html 有关的图形输出，这里只给代码，可以自行运行，查看结果。

注意：提前安装好相应的包。

htmlwidgets 包，这是一个专为 R 语言打造的可视化 JS 库，只需要编写几行 R 语言代码便可生成交互式的可视化页面。目前已经有基于 htmlwidgets 制作的 R 包可供直接调用，具体名称及对应作用见表

R包的名称	功能描述
leaflet	互动地图，与OpenStreetMap, Mapbox, and CartoDB地图互动
dygraphs	时间序列可视化，
plotly	交互式可视化，可以将ggplot2图形转化成交互式的
highcharter	HighchartsJS图形库的R接口
visNetwork	基于vis.js网络可视化
networkD3	基于D3JS网络可视化，
d3heatmap	与D3交互的热图
DT	交互式数据表格
rthreejs	交互式3D图形
rglwidget	提供WebGL场景
DiagrammeR	创建流程图的工具
metricsgraphics	MetricsGraphics.js的htmlwidget接口

4.1 leaflet 包

```
library(leaflet)
leaflet() %>%
  addTiles() %>%
  addMarkers(lng=174.768, lat=-36.852, popup="The birthplace of R")
```

4.2 dygraphs 包

```
library(dygraphs)
lungDeaths <- cbind(mdeaths, fdeaths)
dygraph(lungDeaths)
```

4.3 plotly 包

```
library(plotly)
pal <- RColorBrewer::brewer.pal(nlevels(iris$Species), "Set1")
plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length, color = ~Species,
```

```

  colors = pal, mode = "markers")

p <- ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, colour = Species))+
  scale_color_brewer(palette = "Set1")+
  geom_point()
ggplotly(p)

```

4.4 DT 包

```

library(DT)
datatable(iris)

```

4.5 networkD3 包

```

library(networkD3)

src <- c("A", "A", "A", "A", "B", "B", "C", "C", "D")
target <- c("B", "C", "D", "J", "E", "F", "G", "H", "I")
networkData <- data.frame(src, target)
simpleNetwork(networkData, zoom = T)

data(MisLinks)
data(MisNodes)
forceNetwork(Links = MisLinks, Nodes = MisNodes, Source = "source",
             Target = "target", Value = "value", NodeID = "name",
             Group = "group", opacity = 0.8)

```

4.6 利用 Shiny 包实现可交互的 Web 应用（待补充-更新可见公众号）

shiny 的官网包含了非常多的内容，包括详细教程，案例等。网站地址如下：

<https://shiny.rstudio.com/tutorial/>

Chapter 5

附录

为了方便初学者快速入门以及文稿的完整性, 这里罗列了常用问题的介绍, 包括: 安装 R 和 Rstudio, 使用 Rstudio 可能遇到的问题以及如何获取帮助。

注: 该部分来源于《数据科学中的 R 语言》。

安装 R 和 Rstudio

R 软件是一个自由、开源软件平台, 具有统计分析、可视化和编程的强大功能。你可以从这里免费下载。为了更好的使用 R 软件, 我推荐大家使用 RStudio 这个 IDE。这里有个在线教程帮助我们熟悉 R 和 RStudio。

安装 R

我们从官方网站 <http://cran.r-project.org> 下载, 网站界面感觉有点朴素:

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, Windows and Mac users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

安装 RStudio

安装完 R, 还需要安装 RStudio。有同学可能要问 R 与 RStudio 是什么关系呢? 打个比方吧, R 就像汽车的发动机, RStudio 就是汽车的仪表盘。但我更觉得 R 是有趣的灵魂, 而 Rstudio 是好看的皮囊。

R: Engine**RStudio: Dashboard**

同样，我们从官方网站下载并安装，如果你是苹果系统的用户，选择苹果系统对应的 rstudio 版本即可。

- <https://www.rstudio.com/download>
- 选择 RStudio Desktop

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.456 - Windows Vista/7/8/10	85.8 MB	2018-07-19	24ca3fe0dad8187aab4bfbb9dc2b5ad
RStudio 1.1.456 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-07-19	4fc4f4f70845b142bf96dc1a5b1dc556
RStudio 1.1.456 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-07-19	3493f9d5839e3a3d697f40b7bb1ce961
RStudio 1.1.456 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-07-19	863ae806120358fa0146e4d14cd75be4
RStudio 1.1.456 - Ubuntu 16.04+/Debian 9+ (64-bit)	64.9 MB	2018-07-19	d96e63548c2add890bac633bdb883f32
RStudio 1.1.456 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-07-19	1df56c7cd80e2634f8a9fdd11ca1fb2d
RStudio 1.1.456 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-07-19	5e77094a88fdbddddd0d35708752462

Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 1.1.456 - Windows Vista/7/8/10	122.9 MB	2018-07-19	659d6bfe716d8c97acbe501270d89fa3
RStudio 1.1.456 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	90 MB	2018-07-19	63117c159deca4d01221a8069bd45373
RStudio 1.1.456 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	98.3 MB	2018-07-19	c53c32a71a400c6571e36c573f83dfde
RStudio 1.1.456 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.8 MB	2018-07-19	f4ba2509fb00e30c91414c6821f1c85f
RStudio 1.1.456 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	91.4 MB	2018-07-19	c60db6467421aa86c772227da0945a13



这里有个小小的提示：

- 电脑不要用中文用户名，否则Rstudio会杠上中文用户名
- 尽量安装在非系统盘，比如，可以选择安装在D盘
- 安装路径不要有中文和空格。比如，这样就比较好
 - D:/R
 - D:/Rstudio

可能的问题

- 问题 1：如果下载速度太慢，可以选择国内镜像，

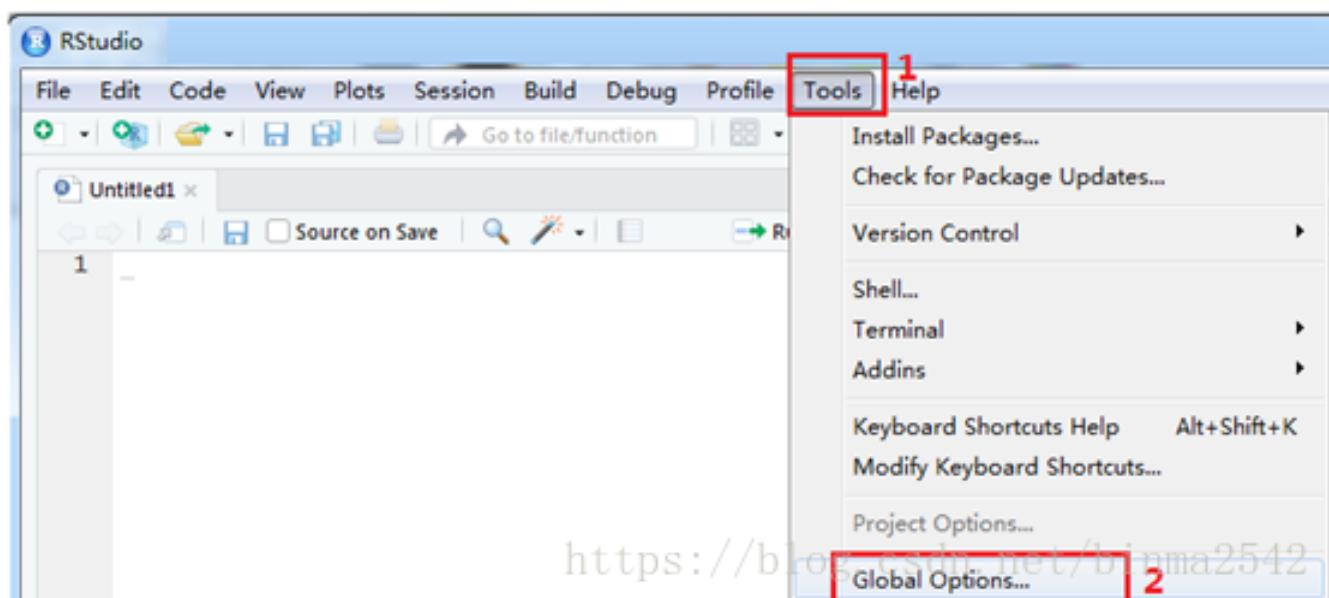


图 5.1: 选择国内镜像 1

然后再输入命令 `install.packages("tidyverse")`，或者直接指定清华大学镜像

```
install.packages("tidyverse", repos = "http://mirrors.tuna.tsinghua.edu.cn/CRAN")
```

- 问题 2：如果遇到如下报错信息

```
Warning in install.packages :
```

```
  unable to access index for repository http://cran.rstudio.com/src/contrib:
  cannot open URL 'http://cran.rstudio.com/src/contrib/PACKAGES'
```

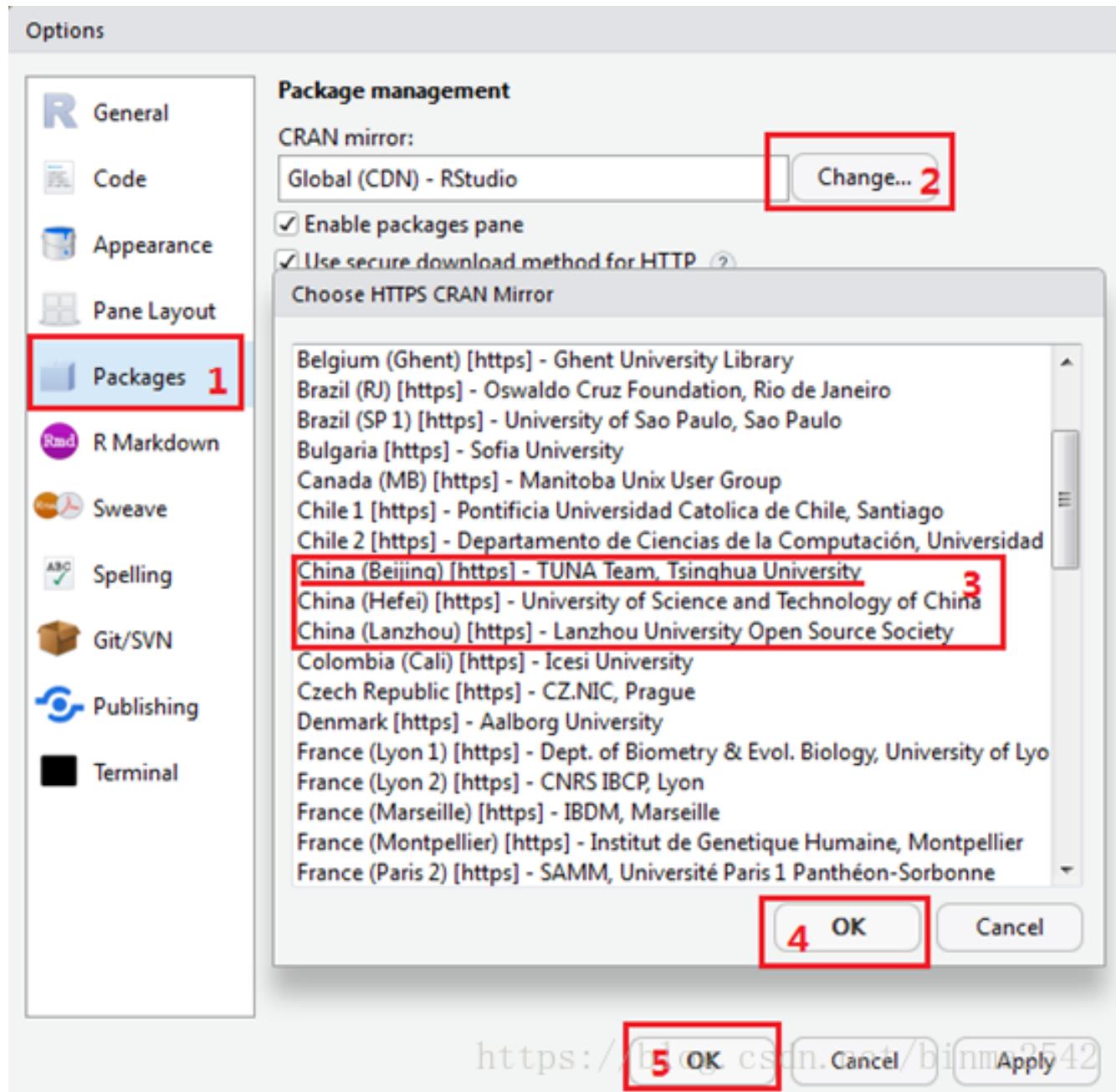


图 5.2: 选择国内镜像 2

输入下面命令后，再试试

```
options(download.file.method="libcurl")
```

或者打开 D:\R\etc\Rprofile.site，添加以下内容：

```
local({r <- getOption("repos")
       r["CRAN"] <- "http://mirrors.tuna.tsinghua.edu.cn/CRAN"
       options(repos=r)})
```

```
options(download.file.method="libcurl")
```

- 问题 3：如果打开代码是乱码，可以试试修改如下设置

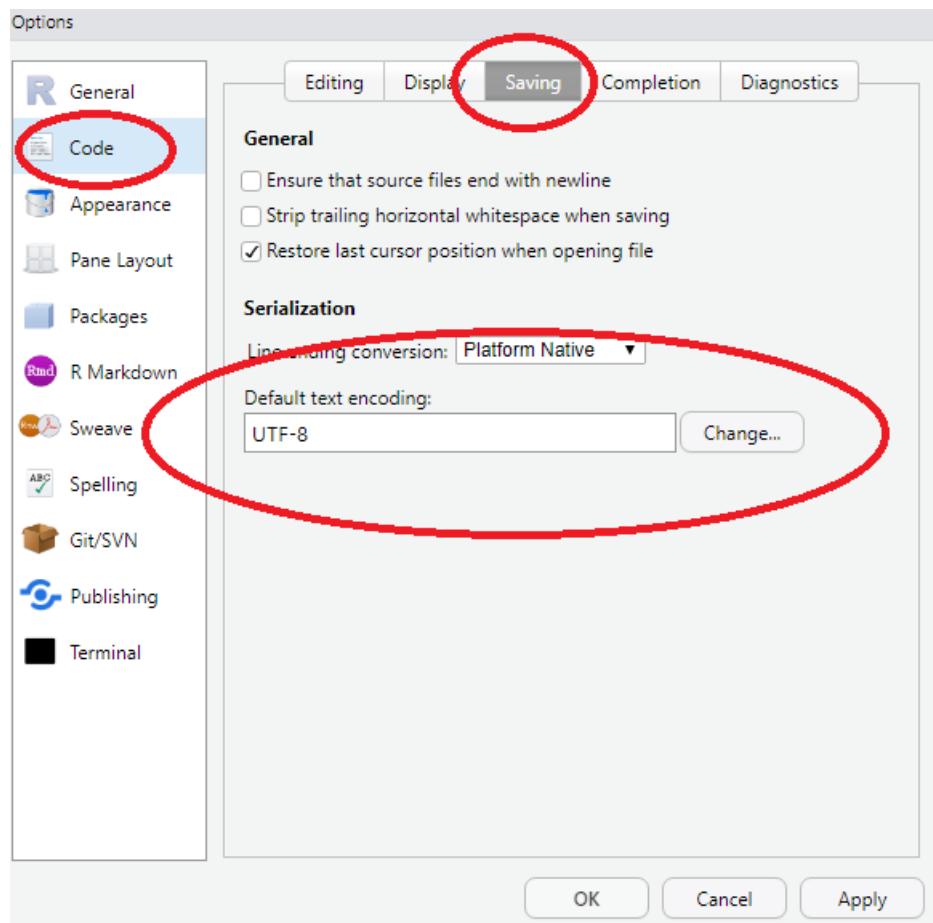


图 5.3: 代码是乱码

- 问题 4：如果每次打开 Rstudio 非常慢，可以在 Rstudio 里将这几个选项取消
- 问题 5：如果 Rstudio 打开是空白

很大的可能是你的电脑用户名是中文的，修改用户名再试试

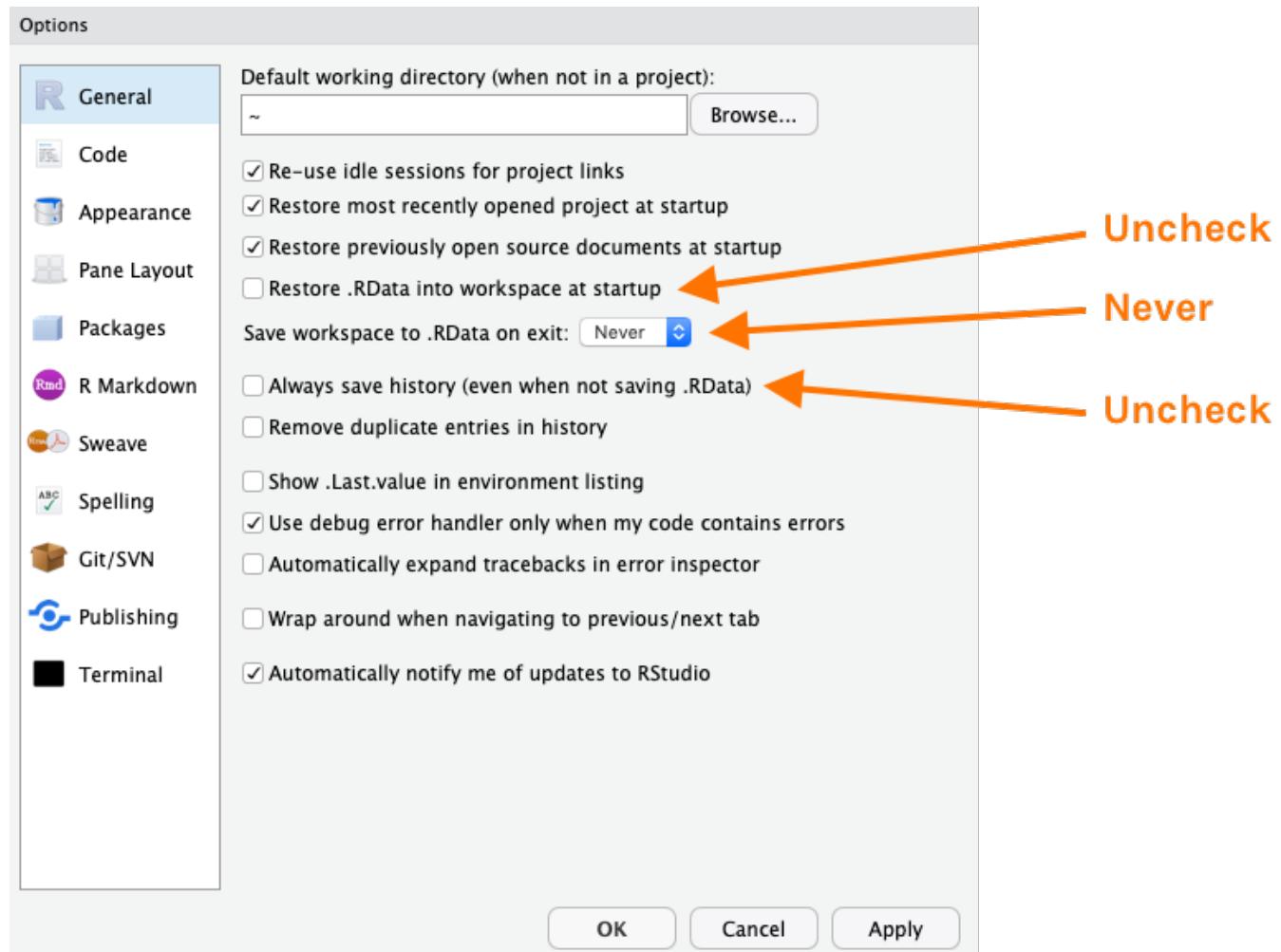


图 5.4: 打开 Rstudio 非常慢

- 问题 6：安装过程中提示，我的系统不能兼容 64 位的 Rstudio。

可能你是低版本的 windows 系统，建议安装旧版本的 Rstudio，可以在这里找到旧版本.

更多 Rstudio 的使用，可参考这里 [introducing-the-rstudio.com](#)

如何获取帮助

- 记住和学习所有的函数几乎是不可能的
- 打开函数的帮助页面 (Rstudio 右下面板的 Help 选项卡)

```
?sqrt
?gather
?spread
?ggplot2
?scale
?map_dfr
```

比如：

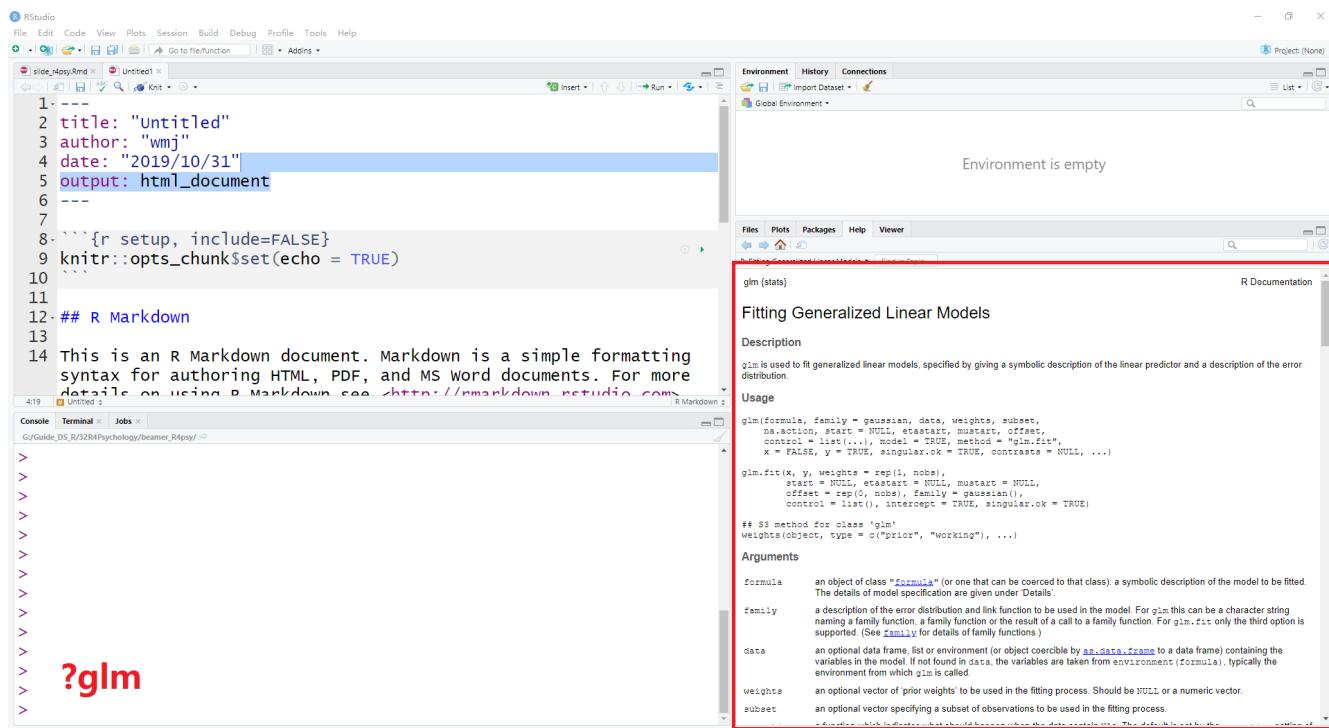


图 5.5: help 帮助

R 语言社区

R 语言社区非常友好，可以在这里找到你问题的答案

- twitter: <https://twitter.com/>
- R-Bloggers: <https://www.r-bloggers.com/>
- kaggle: <https://www.kaggle.com/>
- stackoverflow: <https://stackoverflow.com/questions/tagged/r>
- rstudio: <https://community.rstudio.com/>

Chapter 6

中文图书 Bookdown 模板的基本用法

6.1 安装设置

使用 RStudio 软件完成编辑和转换功能。在 RStudio 中，安装 bookdown 等必要的扩展包。

本模板在安装之前是一个打包的 zip 文件，在适当位置解压（例如，在 C:/myproj 下），得到 MathJax, Books/Cbook, Books/Article 等子目录。本模板在 Books/Cbook 中。

为了利用模板制作自己的中文书，将 Books/Cbook 制作一个副本，改成适当的子目录名，如 Books/Mybook。

打开 RStudio 软件，选菜单“File - New Project - Existing Directory”，选中 Books/Mybook 子目录，确定。这样生成一本书对应的 R project（项目）。

为了将模板内容替换成自己的内容，可以删除文件 0101-usage.Rmd，然后将 1001-chapter01.Rmd 制作几份副本，如 1001-chapter01.Rmd, 2012-chapter02.Rmd, 3012-chapter03.Rmd。各章的次序将按照前面的数值的次序排列。将每个.Rmd 文件内的 {#chapter01}, {#chapter02-sec01} 修改能够反映章节内容的标签文本。所有的标签都不允许重复。参见本模板中的 0101-usage.Rmd 文件。

后面的 §6.3.1 和 §6.3.2 给出了将当前的书转换为网页和 PDF 的命令，复制粘贴这些命令到 RStudio 命令行可以进行转换。

6.2 编写自己的内容

6.2.1 文档结构

除了 index.Rmd 以外，每个.Rmd 文件是书的一章。每章的第一行是用一个井号 (#) 引入的章标题。节标题用两个井号开始，小节标题用三个井号开始。标题后面都有大括号内以井号开头的标签，标签仅用英文大小写字母和减号。

6.2.2 图形自动编号

用 R 代码段生成的图形，只要具有代码段标签，且提供代码段选项 `fig.cap=" 图形的说明文字"`，就可以对图形自动编号，并且可以用如`\@ref(fig:label)` 的格式引用图形。如：

```
plot(1:10, main=" 程序生成的测试图形")
```

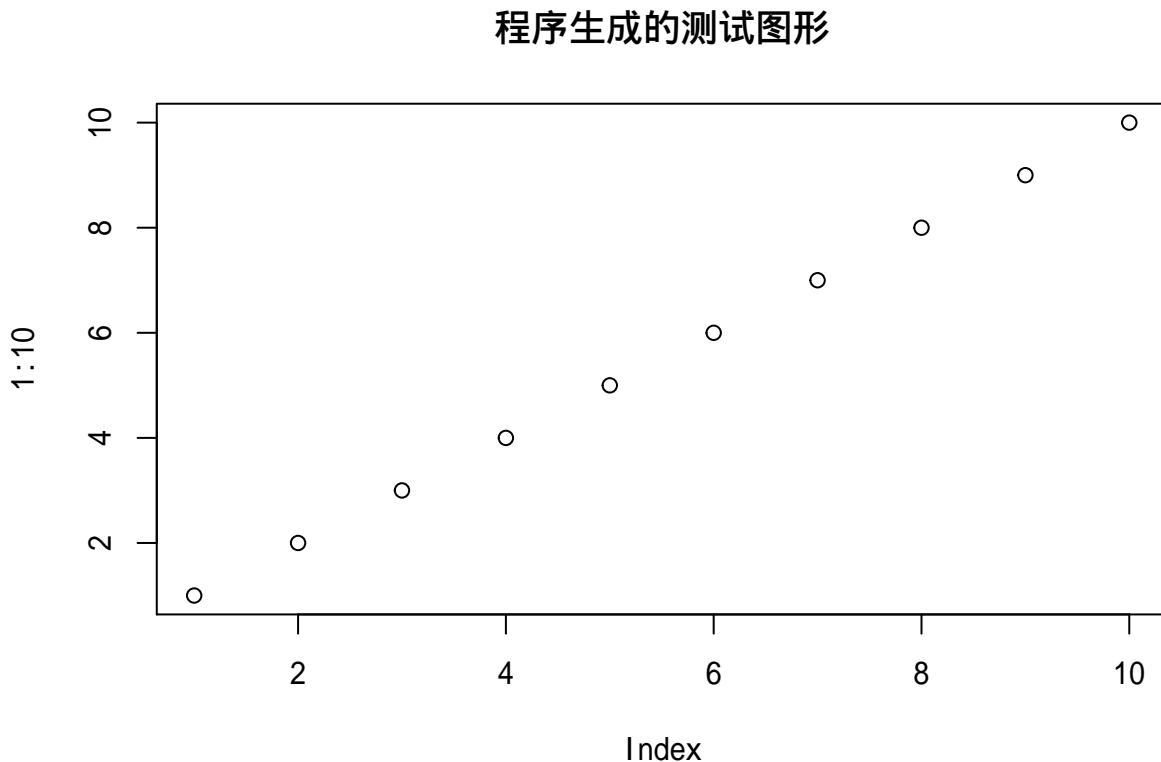


图 6.1: 图形说明文字

引用如：参见图6.1。引用中的 `fig:` 是必须的。

在通过 LaTeX 转换的 PDF 结果中，这样图形是浮动的。

6.2.3 表格自动编号

用 R 代码 `knitr::kable()` 生成的表格，只要具有代码段标签，并且在 `knitr::kable()` 调用时加选项 `caption=" 表格的说明文字"`，就可以对表格自动编号，并且可以用如`\@ref(tab:label)` 的格式引用表格。如：

```
d <- data.frame(" 自变量"=1:10, " 因变量"=(1:10)^2)
knitr::kable(d, caption=" 表格说明文字")
```

引用如：参见表6.1。引用中的 `tab:` 是必须的。

在通过 LaTeX 转换的 PDF 结果中，这样的表格是浮动的。

表 6.1: 表格说明文字

自变量	因变量
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

6.2.4 数学公式编号

不需要编号的公式，仍可以按照一般的 Rmd 文件中公式的做法。需要编号的公式，直接写在 `\begin{align}` 和 `\end{align}` 之间，不需要编号的行在末尾用 `\nonumber` 标注。需要编号的行用 `(\#eq:mylabel)` 添加自定义标签，如

$$\begin{aligned} \Sigma &= (\sigma_{ij})_{n \times n} \\ &= E[(X - \mu)(X - \mu)^T] \end{aligned} \tag{6.1}$$

引用如：协方差定义见(6.1)。

6.2.5 文献引用与文献列表

将所有文献用 bib 格式保存为一个 .bib 文献库，如模板中的样例文件 `mybib.bib`。可以用 JabRef 软件来管理这样的文献库，许多其它软件都可以输出这样格式的文件库。

为了引用某一本书，用如：参见 (Wichmann & Hill, 1982)。

被引用的文献将出现在一章末尾以及全书的末尾，对 PDF 输出则仅出现在全书末尾。

6.3 转换

6.3.1 转换为网页

用如下命令将整本书转换成一个每章为一个页面的网站，称为 gitbook 格式：

```
bookdown::render_book("index.Rmd",
  output_format="bookdown::gitbook", encoding="UTF-8")
```

为查看结果，在 `_book` 子目录中双击其中的 `index.html` 文件，就可以在网络浏览器中查看转换的结果。重新编译后应该点击“刷新”图标。

在章节和内容较多时，通常不希望每次小修改之后重新编译整本书，这时类似如下的命令可以仅编译一章，可以节省时间，缺点是导航目录会变得不准确。命令如：

```
bookdown::preview_chapter("1001-chapter01.Rmd",
  output_format="bookdown::gitbook", encoding="UTF-8")
```

单章的网页可以通过网络浏览器中的“打印”功能，选择一个打印到 PDF 的打印机，可以将单章转换成 PDF 格式。

6.3.2 生成 PDF

如果想将 R Markdown 文件借助于 LaTeX 格式转换为 PDF，需要在系统中安装一个 TeX 编译器。现在的 rmarkdown 包要求使用 tinytex 扩展包以及配套的 TinyTeX 软件包，好像不再支持使用本机原有的 LaTeX 编译系统，如果不安装 tinytex，编译为 PDF 格式时会出错。TinyTeX 优点是直接用 R 命令就可以安装，更新也由 R 自动进行，不需要用户干预。但是，安装时需要从国外网站下载许多文件，有因为网络不畅通而安装失败的危险。

为了安装 R 的 tinytex 扩展包和单独的 TinyTeX 编译软件，应运行：

```
install.packages('tinytex')
tinytex::install_tinytex()
```

安装过程需要从国外的服务器下载许多文件，在国内的网络环境下有可能因为网络超时而失败。如果安装成功，TinyTeX 软件包在 MS Windows 系统中一般会安装在 `C:\Users\用户名\AppData\Roaming\MikTeX` 目录中，其中“用户名”应替换成系统当前用户名。如果需要删除 TinyTeX 软件包，只要直接删除那个子目录就可以。

为了判断 TinyTeX 是否安装成功，在 RStudio 中运行

```
tinytex::is_tinytex()
```

结果应为 TRUE，出错或者结果为 FALSE 都说明安装不成功。在编译 `pdf_book` 时，可能会需要联网下载 LaTeX 所需的格式文件。

Bookdown 借助操作系统中安装的 LaTeX 编译软件 TinyTeX 将整本书转换成一个 PDF 文件，这需要用户对 LaTeX 有一定的了解，否则一旦出错，就完全不知道如何解决。用户如果需要进行 LaTeX 定制，可修改模板中的 `preamble.tex` 文件。

转换为 PDF 的命令如下：

```
bookdown::render_book("index.Rmd",
  output_format="bookdown::pdf_book", encoding="UTF-8")
```

在 `_book` 子目录中找到 `CBook.pdf` 文件，这是转换的结果。`CBook.tex` 是作为中间结果的 LaTeX 文件，如果出错可以从这里查找错误原因。

转换 PDF 对于内容多的书比较耗时，不要过于频繁地转换 PDF，在修改书的内容时，多用 `bookdown::preview_chapter` 和转换为 gitbook 的办法检验结果。定期地进行转换 PDF 的测试。每增加一章后都应该试着转换成 PDF 看有没有错误。命令如：

```
bookdown::preview_chapter("1001-chapter01.Rmd",
  output_format="bookdown::gitbook", encoding="UTF-8")
```

6.3.3 上传到网站

如果书里面没有数学公式，则上传到网站就只要将 `_book` 子目录整个地用 ftp 软件传送到自己的网站主目录下的某个子目录即可。但是，为了支持数学公式，就需要进行如下的目录结构设置：

1. 设自己的网站服务器目录为 `/home/abc`，将 MathJax 目录上传到这个目录中。
2. 在 `/home/abc` 中建立新目录 `Books/Mybook`。
3. 将 `_book` 子目录上传到 `/home/abc/Books/Mybook` 中。
4. 这时网站链接可能类似于 `http://dept.univ.edu.cn/~abc/Books/Mybooks/_book/index.html`，具体链接地址依赖于服务器名称与主页所在的主目录名称。

如果有多本书，MathJax 仅需要上传一次。因为 MathJax 有三万多个文件，所以上传 MathJax 会花费很长时间。

Chapter 7

参考书籍

这里书籍都有线上免费版本，可以点击书籍名称跳转。同时感谢西京学院刘琦老师对文稿提供的帮助。

- R 语言基础教程——李东风
- 数据科学中的 R 语言——王敏杰
- ggplot2 书
- ggplot2 画廊
- R 数据科学

其他有关 **R** 语言的书籍可在庄闪闪的 **R** 语言手册(可跳转) 命令窗口输入 “**R**” 获取。也欢迎关注我的个人公众号，和我一起学 R，统计和数据科学。



图 7.1: 个人公众号: 庄闪闪的 R 语言手册

The screenshot shows the WeChat public account interface for '庄闪闪的R语言手册'. At the top, there are three recent posts:

- 庄闪闪的R语言手册 ★**: A post from 3:59 PM, featuring a profile picture of a person with short hair and a blue background. It has 71 original contents and 225 friends following.
- R语言学习**: A post from 3:59 PM, featuring a profile picture of a person with short hair and a blue background. It has 1 original content and 1 friend following.
- R语言数据可视化**: A post from 3:59 PM, featuring a profile picture of a person with short hair and a blue background. It has 1 original content and 1 friend following.

Below these are sections for **可视化** (Visualization) and **数据科学** (Data Science). The **可视化** section includes a video thumbnail for 'Rstudio 界面介绍 (上)' (Introduction to Rstudio Interface (Part 1)) uploaded by the author at 11:48 PM yesterday. The **数据科学** section includes a post titled '2020总结, 2021展望' (Summary of 2020, Outlook for 2021) with a blue background and white text.

The main content area displays a grid of articles under the **R包** (R packages) category, each with a thumbnail image and a brief description:

- ggridges包—峰峦图详细介绍**: Original content by the author.
- esquisse包—不写代码生成ggplot图**: Original content by the author.
- calendR包—私人定制专属日历**: Original content by the author.
- corrplot包：相关性矩阵可视化**: Original content by the author.
- R可视化 | 三元相图**: Original content by the author.
- R可视化|2021年日历大派送**: Original content by the author.
- R可视化|瀑布图**: Original content by the author.
- R可视化|等高线图**: Original content by the author.

At the bottom, there is a footer note: '小编第一次录视频，多多见谅，以后继续改进。' (The editor's first video recording, please forgive, I will continue to improve.)

其他联系方式 (可跳转):

- 知乎
- Github
- CSDN
- b 站

Bibliography

Wichmann, B. A., & Hill, I. D. (1982). Algorithm as 183: An efficient and portable pseudo-random number generator. *Applied Statistics*, 31, 188–190. Remarks: 34, 198 and 35, 89.