

# 庄闪闪的可视化手册

庄闪闪

22/7/2022



# 目录

简介	7
<b>I R 语言基础</b>	<b>9</b>
<b>1 Base 系列</b>	<b>11</b>
<b>2 Tidyverse 系列</b>	<b>13</b>
<b>II R 语言可视化</b>	<b>15</b>
<b>3 使用基础包绘图</b>	<b>17</b>
3.1 绘制基本图形 . . . . .	17
3.2 修改图形参数 . . . . .	34
3.3 绘制组合图形 . . . . .	48
3.4 保存图形 . . . . .	51
<b>4 使用 ggplot2 包绘图</b>	<b>55</b>
4.1 简介 . . . . .	55
4.2 qplot . . . . .	55
4.3 ggplot2 包图形语法 . . . . .	61
4.4 几何对象 . . . . .	70
4.5 统计变换 . . . . .	72
4.6 刻度 scale . . . . .	74
4.7 坐标系 . . . . .	81
4.8 分面 . . . . .	84
4.9 标题、标注、指南、拼接 . . . . .	88
4.10 保存图片 . . . . .	94
4.11 例子 . . . . .	95

<b>5 常用图形</b>	<b>99</b>
5.1 散点图 . . . . .	103
5.2 气泡图 . . . . .	108
5.3 线性图 . . . . .	109
5.4 相关系数图/热力图 . . . . .	110
5.5 条形图 . . . . .	112
5.6 箱线图 . . . . .	116
5.7 小提琴图 . . . . .	118
5.8 饼图 . . . . .	120
5.9 冲积图 . . . . .	121
5.10 系谱图 . . . . .	122
5.11 哑铃图 . . . . .	123
5.12 斜率图 . . . . .	124
5.13 堆叠面积图 . . . . .	125
5.14 棒棒糖图 . . . . .	126
5.15 树状图 . . . . .	128
5.16 瀑布图 . . . . .	129
5.17 双标图 . . . . .	130
5.18 雷达图/星图/蜘蛛图 . . . . .	133
<b>6 其他相关拓展包</b>	<b>135</b>
6.1 ggplot2 官网 109 种拓展包 . . . . .	135
6.2 整理的相关扩展包 . . . . .	138
<b>7 认识交互式绘图工具</b>	<b>139</b>
7.1 leaflet 包 . . . . .	140
7.2 dygraphs 包 . . . . .	140
7.3 plotly 包 . . . . .	140
7.4 DT 包 . . . . .	141
7.5 networkD3 包 . . . . .	141
7.6 利用 Shiny 包实现可交互的 Web 应用 . . . . .	141
<b>8 常用案例</b>	<b>143</b>
8.1 绘制不同分布的 QQ 图 . . . . .	143
8.2 绘制混合密度函数图以及添加分位数线 . . . . .	148
8.3 绘制不同参数的密度函数 . . . . .	151
8.4 散点图中加入第三变量的密度函数 . . . . .	153
8.5 绘制 Gamma 分布相关图形 . . . . .	154
8.6 纯手动复现密度图 . . . . .	158

8.7 绘制贝叶斯分析相关图形 . . . . .	161
<b>III R 语言文档沟通</b>	<b>171</b>
<b>9 R Markdown 入门教程</b>	<b>173</b>
9.1 Rmarkdown 简介 . . . . .	173
9.2 Rmarkdown 流程演示 . . . . .	174
9.3 Rmarkdown 的主题格式 . . . . .	194
9.4 总结一些常用技巧 . . . . .	201
<b>10 其他方式的文档沟通</b>	<b>213</b>
10.1 Typora . . . . .	213
10.2 xaringan 幻灯片 . . . . .	213
10.3 制作幻灯片 . . . . .	213
10.4 Rbeamer . . . . .	214
10.5 Dashboards . . . . .	214
10.6 shiny . . . . .	214
10.7 blogdown . . . . .	214
10.8 制作简历 . . . . .	214
10.9 bookdown . . . . .	215
10.10 小编有话说 . . . . .	215
<b>11 附录</b>	<b>217</b>
安装 R 和 Rstudio . . . . .	217
可能的问题 . . . . .	219
如何获取帮助 . . . . .	223
R 语言社区 . . . . .	224
<b>12 参考书籍</b>	<b>225</b>



# 简介

这本电子书是《庄闪闪的可视化笔记》。主要内容来自一次市场调查大赛公益课程的教学内容。

这个文稿主要是在一次公益课上整理出来的。经过一年的时候，小编在实践中添加了一些新的内容，希望能为各位科研人提供一些参考。对应的上课视频如下：[R 语言可视化教程](#)。

笔记还会持续更新，读者可以通过我的公众号《庄闪闪的 R 语言手册》了解最新的学习笔记。

我是庄闪闪，这是我的个人博客，如果有什么问题，可以通过以下方式联系我：

[GitHub](#)、[B 站](#)、[知乎](#)、[CSDN](#)、[小红书](#)、[掘金](#)



## Part I

# R 语言基础



# Chapter 1

## Base 系列



## Chapter 2

### Tidyverse 系列



## Part II

# R 语言可视化



# Chapter 3

## 使用基础包绘图

### 3.1 绘制基本图形

#### 3.1.1 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。

##### 3.1.1.1 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。

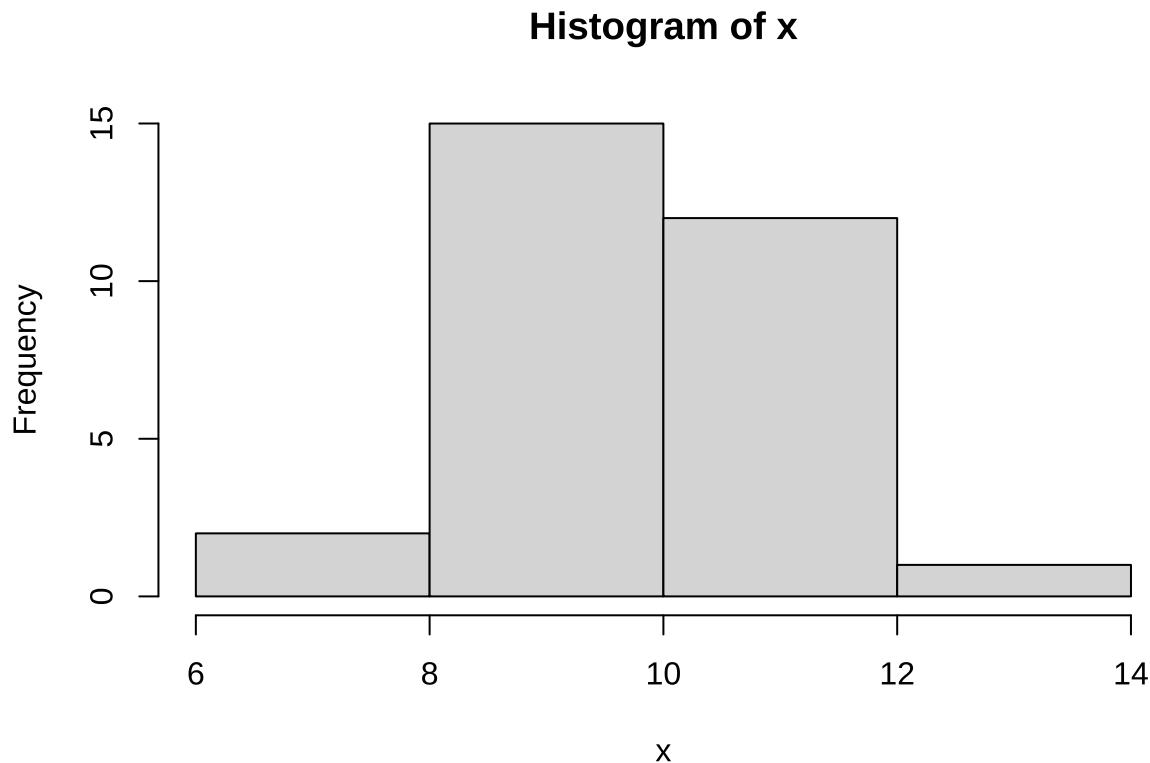
例子：我们使用模拟数据进行讲解，通过正态分布产生 30 个数据。

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

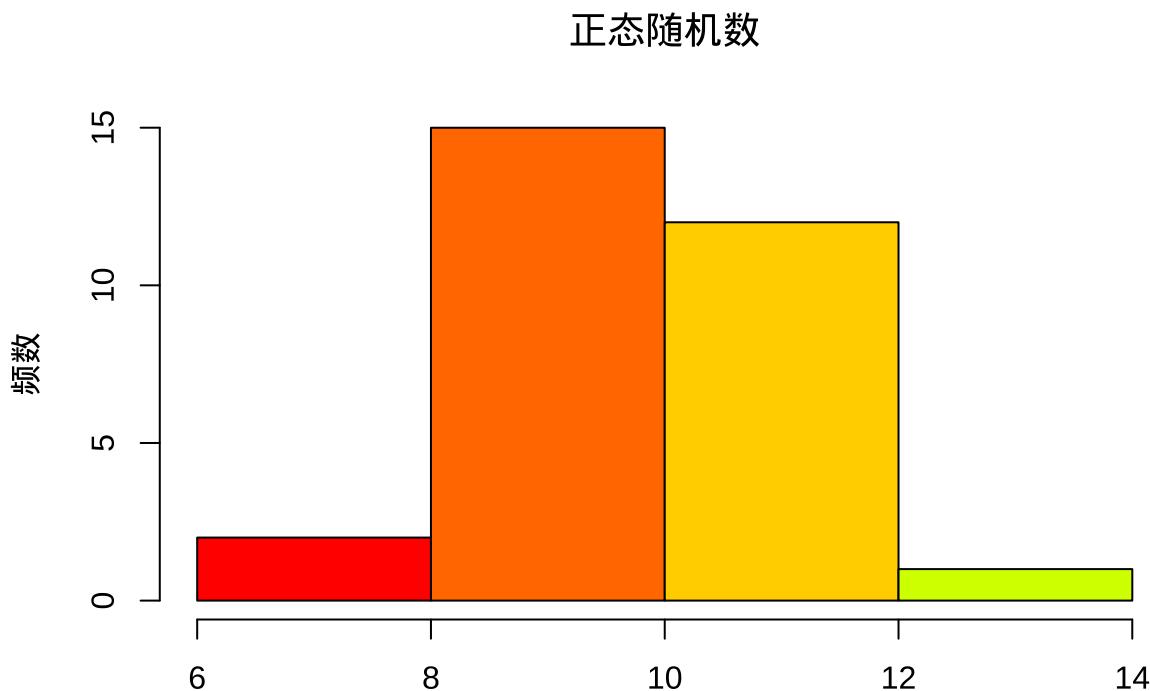
## [1] 10.70 10.61  9.75  9.63  9.25  9.06 10.87  9.46 10.67  9.07 10.80 11.44
## [13]  9.71 12.32  9.75 11.49  7.78  8.94  9.60 10.38  9.37  9.94 10.26  9.39
## [25]  7.78 10.72 10.82  8.19  9.19 10.03
```

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)
```



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

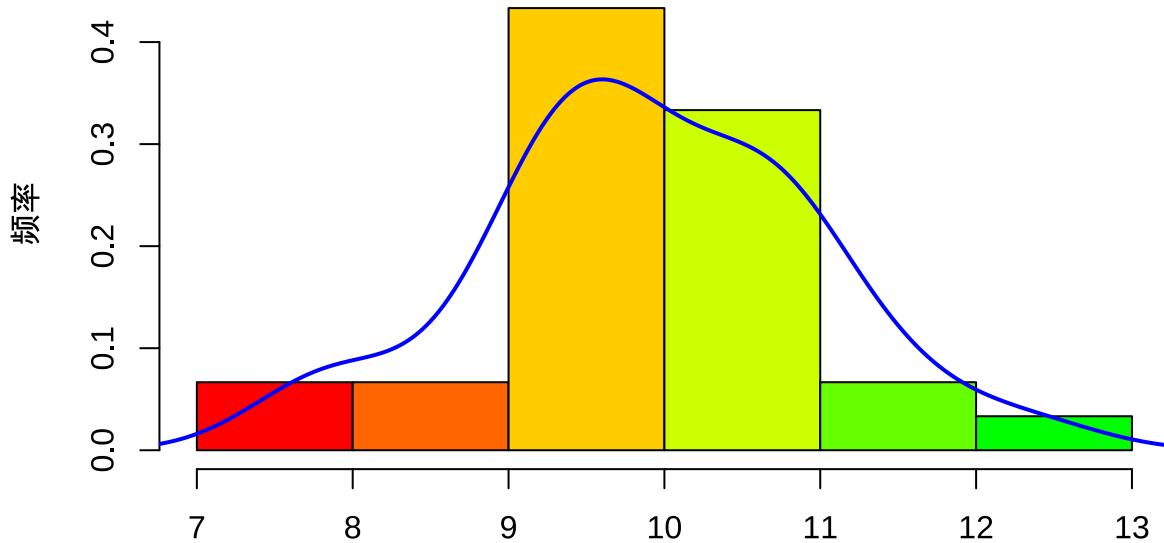


`breaks = 3` 表示 x 轴分为 3 个节点。其他设置可参考帮助文档，即`?hist`。这里加入其他参数 `col`, `main`, `xlab`, `ylab`, 分别表示颜色, 主题名称, x 轴名称, y 轴名称设置。细节将会在下面一章进行详细解释。

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')
```

正态随机数

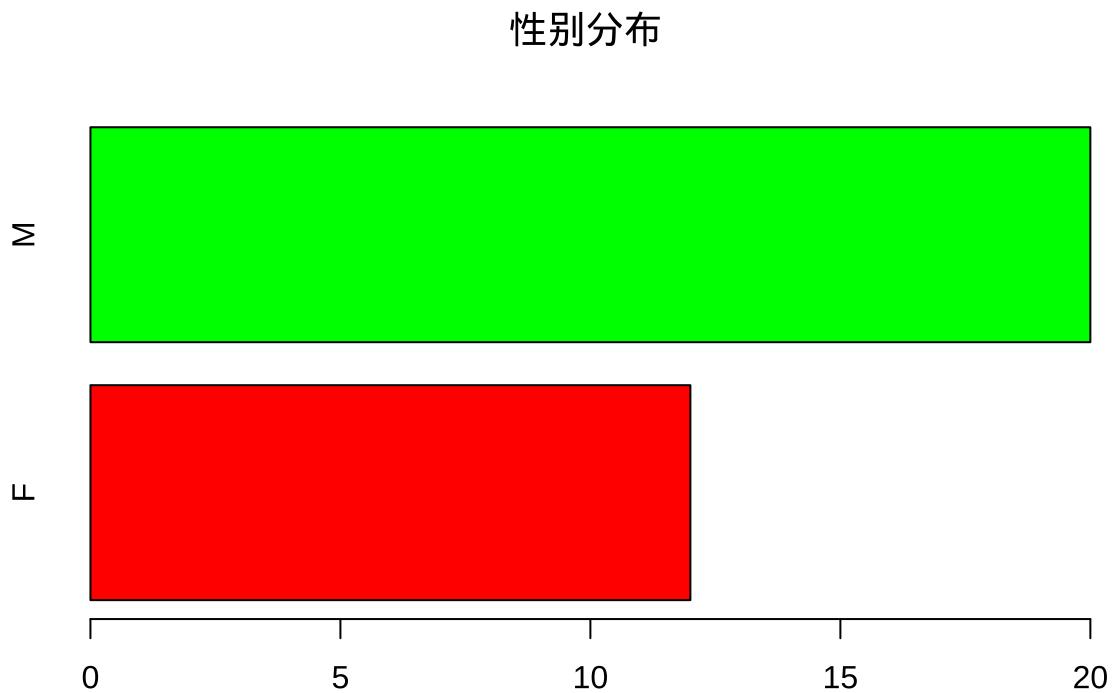


### 3.1.1.2 条形图

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)
```



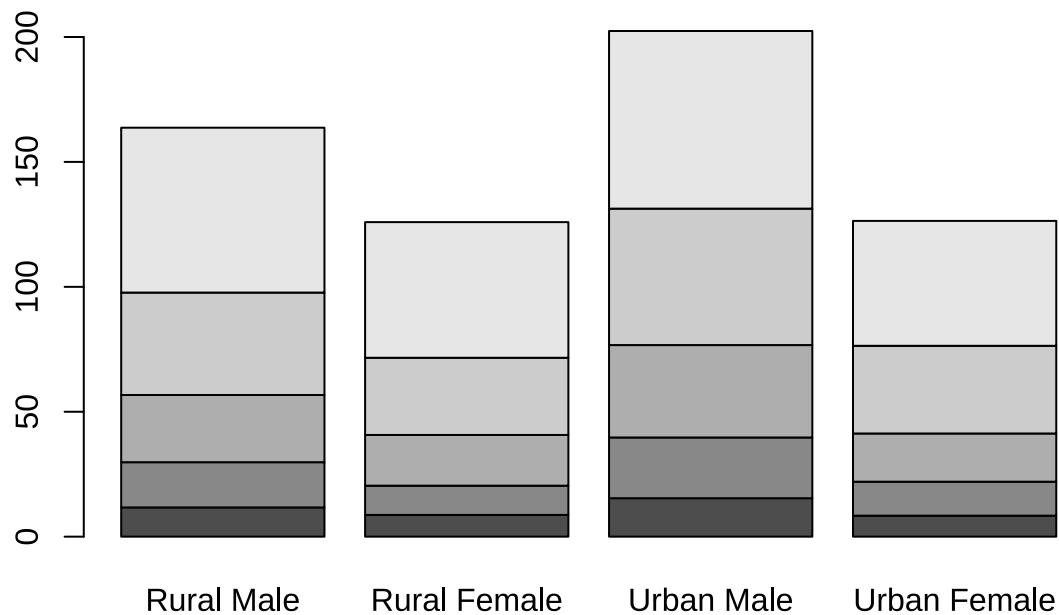
数据介绍: VADeaths 数据集记录的是 1940 年 Viginia(弗吉尼亚洲) 不同人群 (Rural Male、Rural Female、Urban Male、Urban Female) 中每一千人的死亡情况。

例子: 数据前 6 行展示如下:

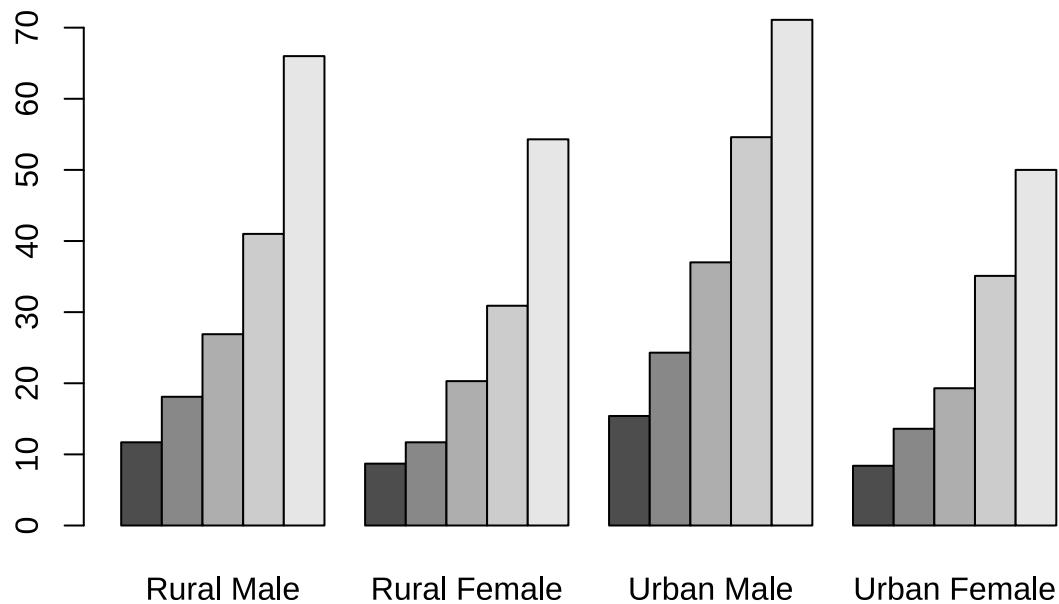
	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

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)
```



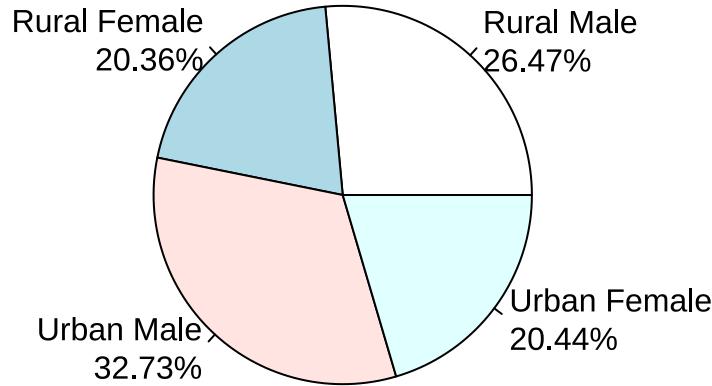
结论：随着年龄的增长，Virginia 人群的死亡率逐渐增加，并且在 4 类人群中，Urban Male 的死亡率比同年龄段的其他群体的死亡率高。同时，在同一环境下，相同年龄段的男性的死亡率要比女性高。

### 3.1.1.3 饼图

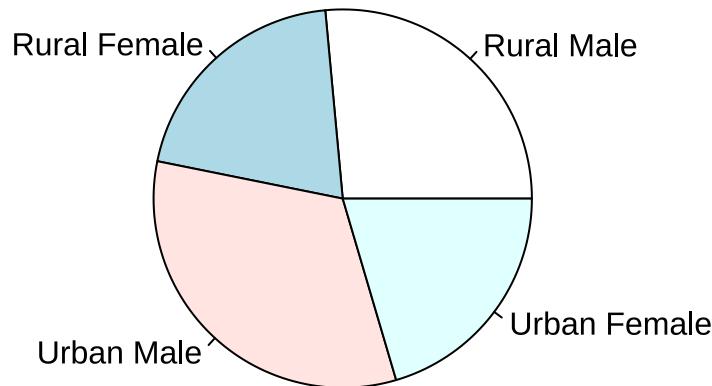
概念介绍：将各项的大小与各项总和的比例。反映部分与部分、部分与整体之间的比例关系。

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。

### 3.1.1.4 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。

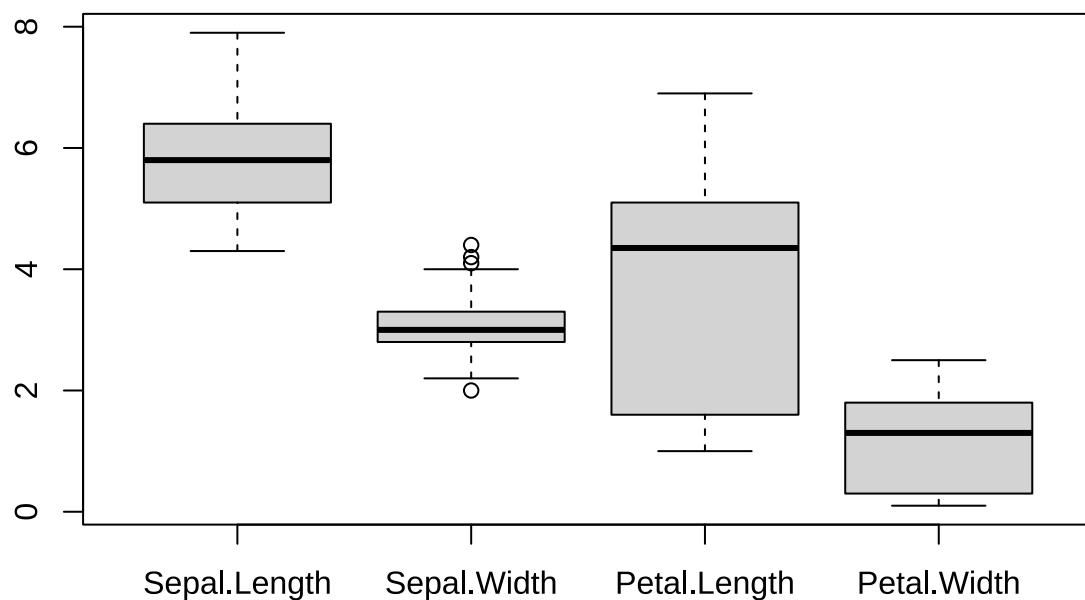
前 6 行数据如下：

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。

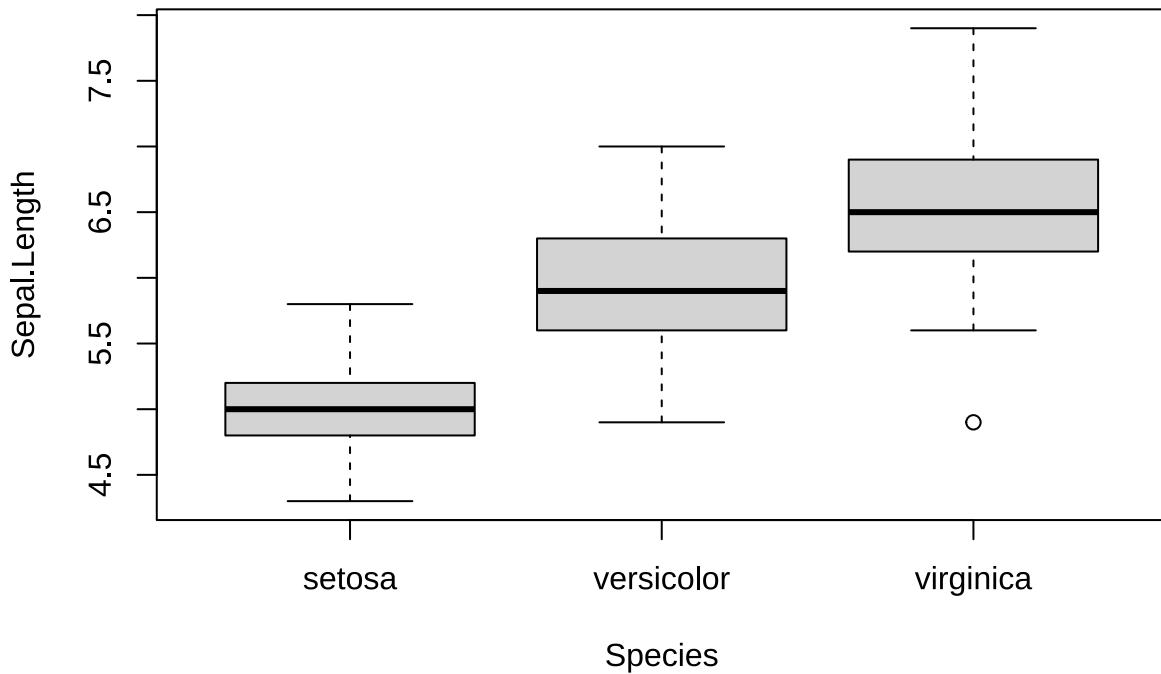
```
attach(iris)
boxplot(iris[1:4], main = '单独的箱线图')
```

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。

### 3.1.2 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。

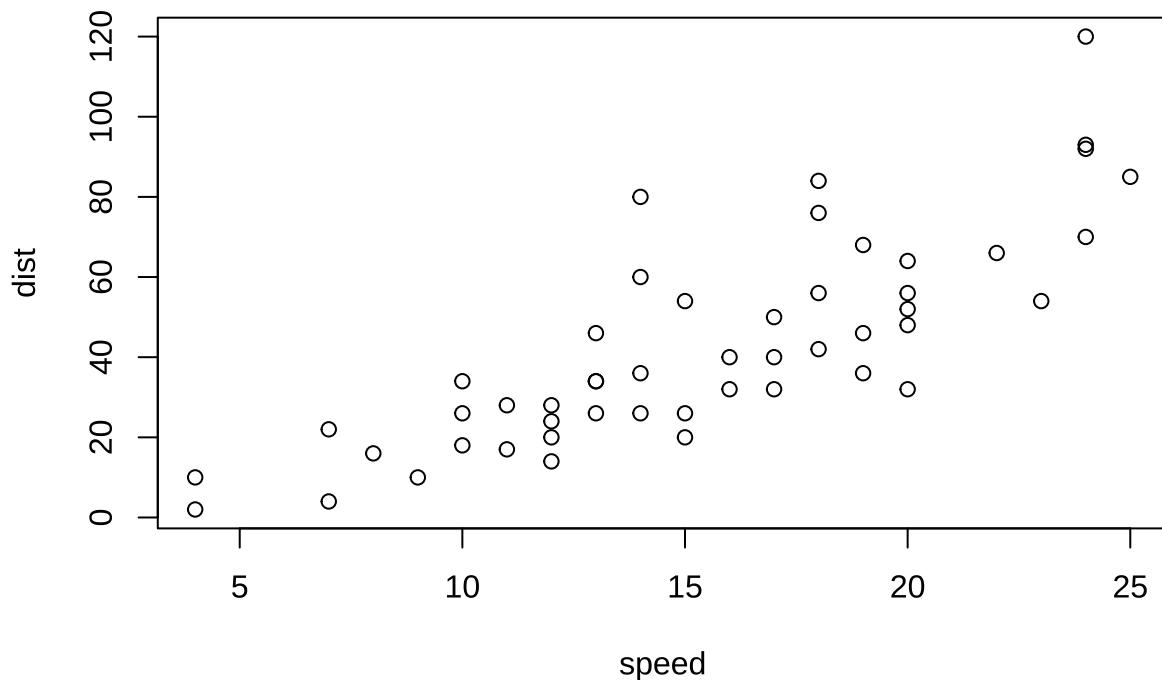
#### 3.1.2.1 散点图

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。

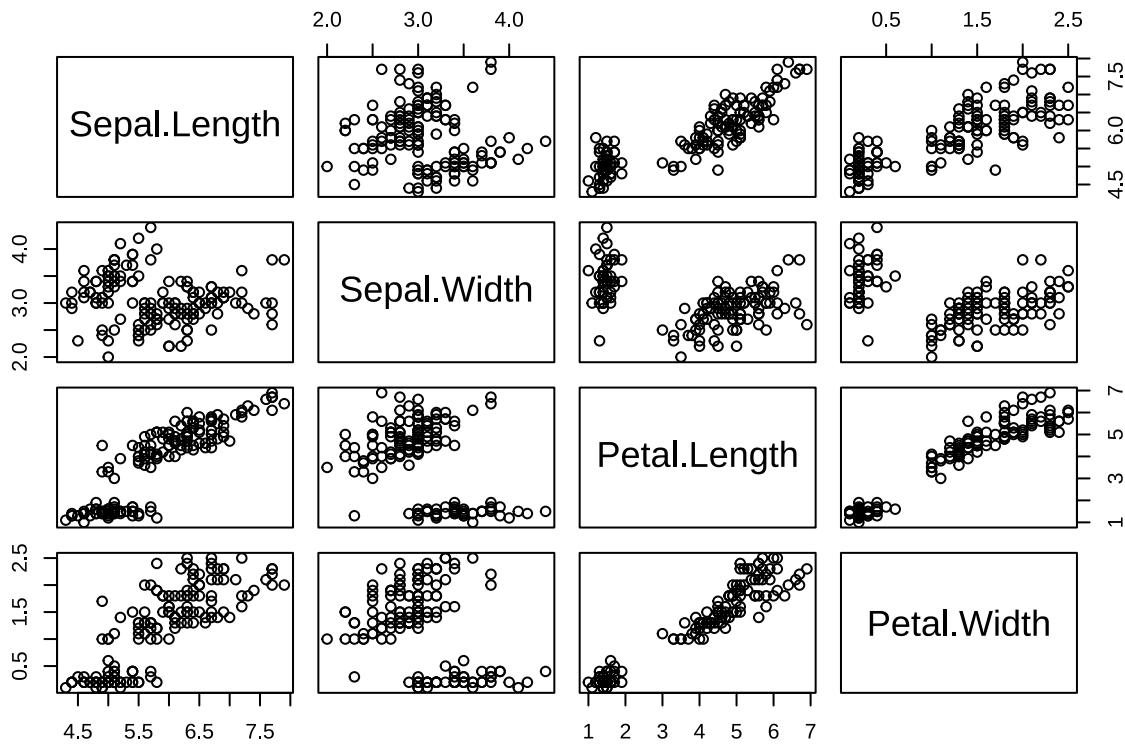
### 3.1.2.2 散点矩阵图

**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])
```

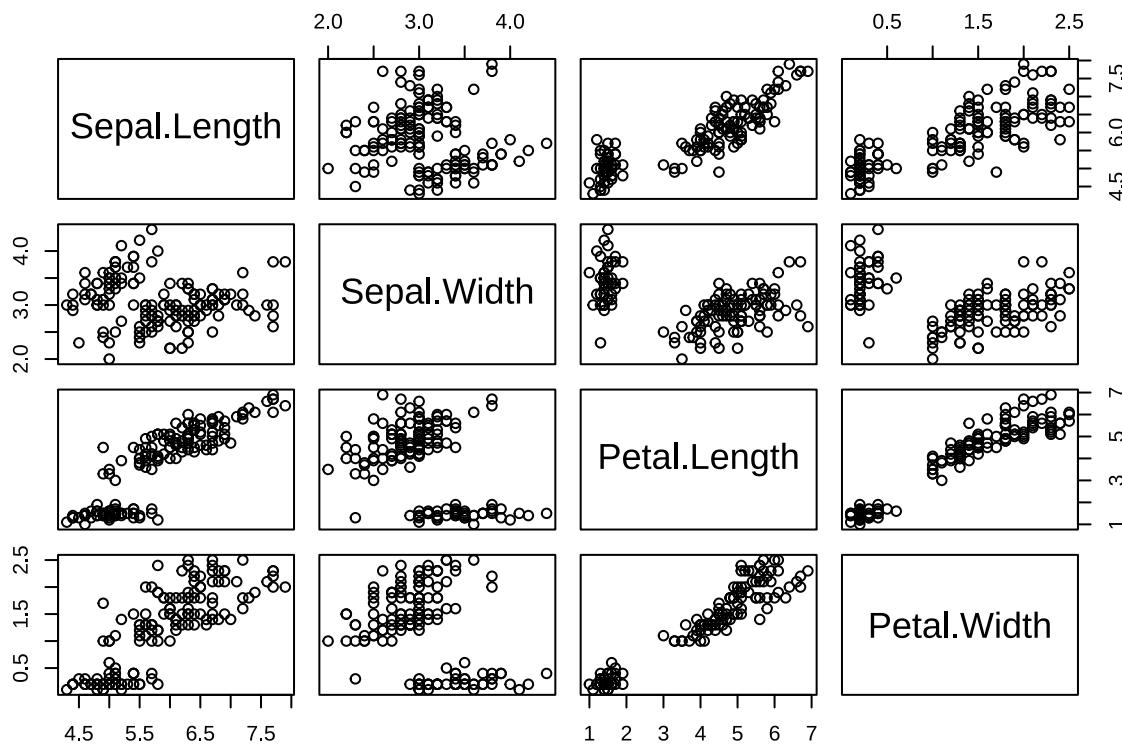


结论：花瓣长度（Petal.length）与花瓣宽度（Petal.Width）有明显的线性关系，其余属性之间的关系不是很明显。

- 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
```



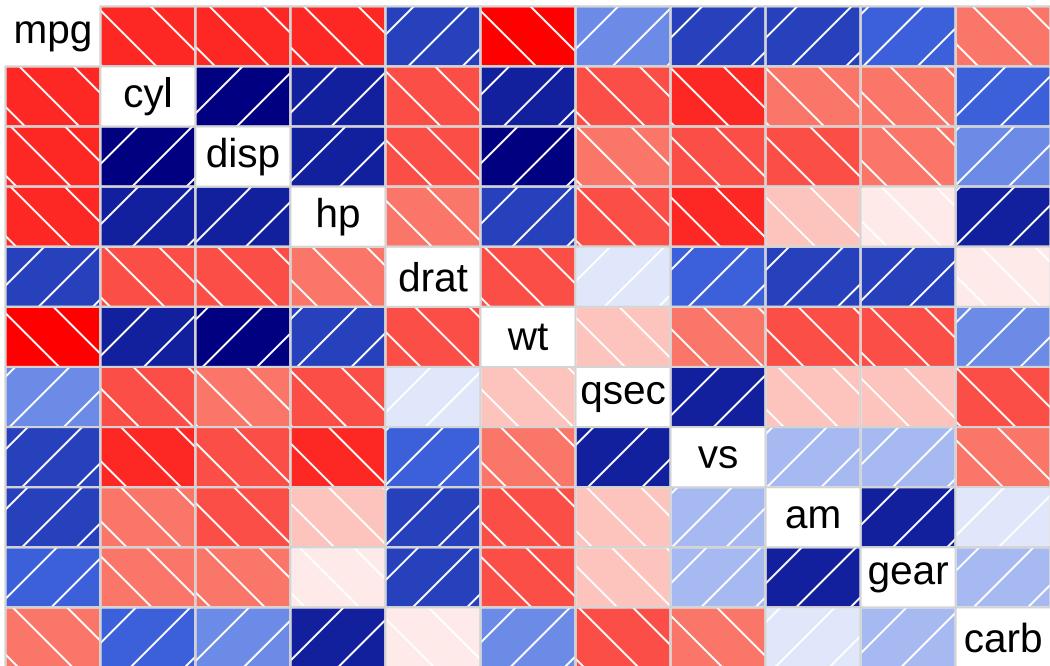
### 3.1.2.3 多变量相关矩阵图

**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。

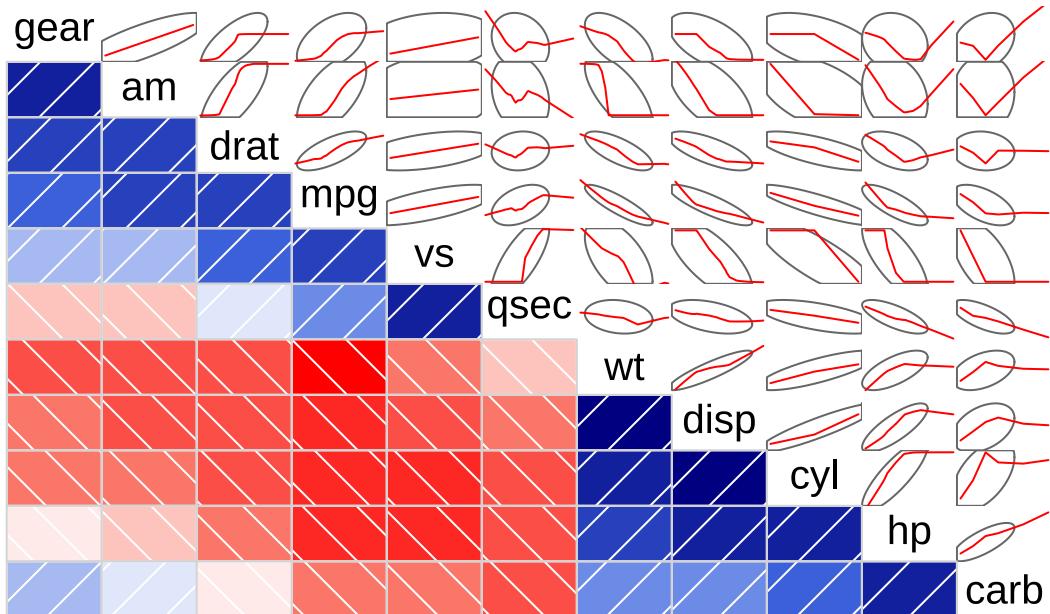
**例子：**下面、根据这个数据集，绘制适用中不同元素描述相关性大小的图。

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



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

**Correlogram of mtcars intercorrelations**



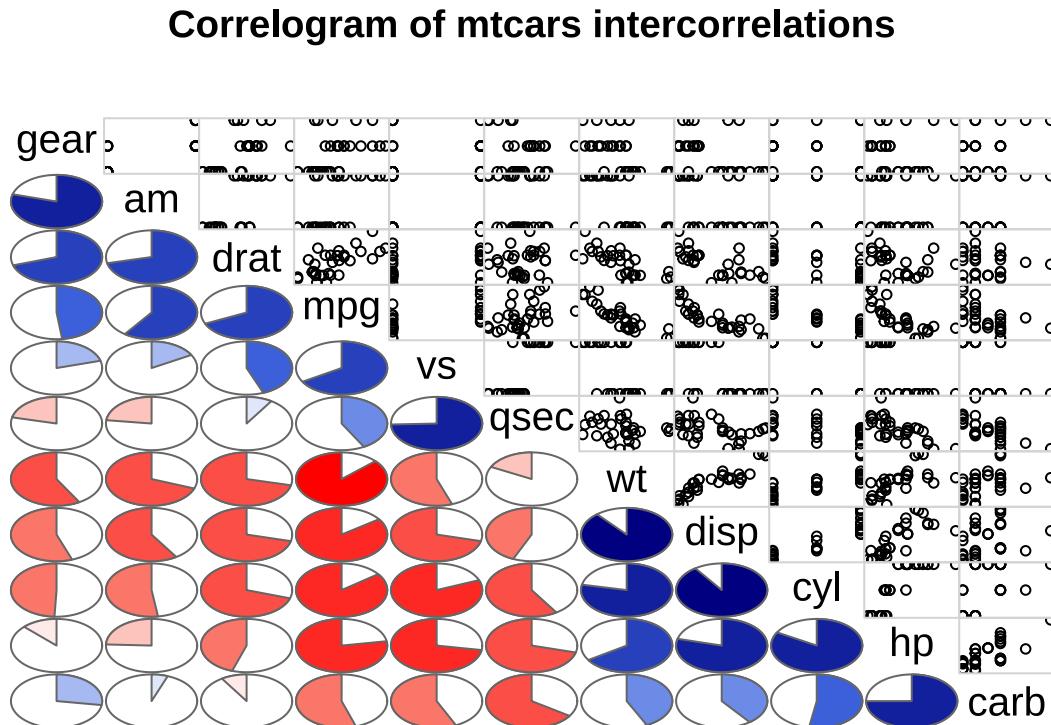
```
# 相关图，主对角线上方绘制置信椭圆和平滑拟合曲线，主对角线下方绘制阴影
```

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")
```

## 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)	-0.87 (-0.93,-0.74)	-0.85 (-0.92,0.71)	-0.85 (-0.93,-0.72)	-0.78 (-0.89,-0.59)	-0.55 (-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)	-0.66 (-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	0.89	disp	0.90 (0.81,0.95)	0.79 (0.61,0.89)	0.39 (0.05,0.55)
-0.49	-0.52	-0.70	-0.85	-0.81	-0.59	0.78	0.90	cyl	0.83 (0.68,0.92)	0.53 (0.22,0.74)
-0.13	-0.24	-0.45	-0.78	-0.72	-0.71	0.66	0.79	0.83	hp	0.75 (0.51,0.87)
0.27	0.06	-0.09	-0.55	-0.57	-0.66	0.43	0.39	0.53	0.75	carb

### 3.1.3 绘制其他图形

#### 3.1.3.1 核密度图

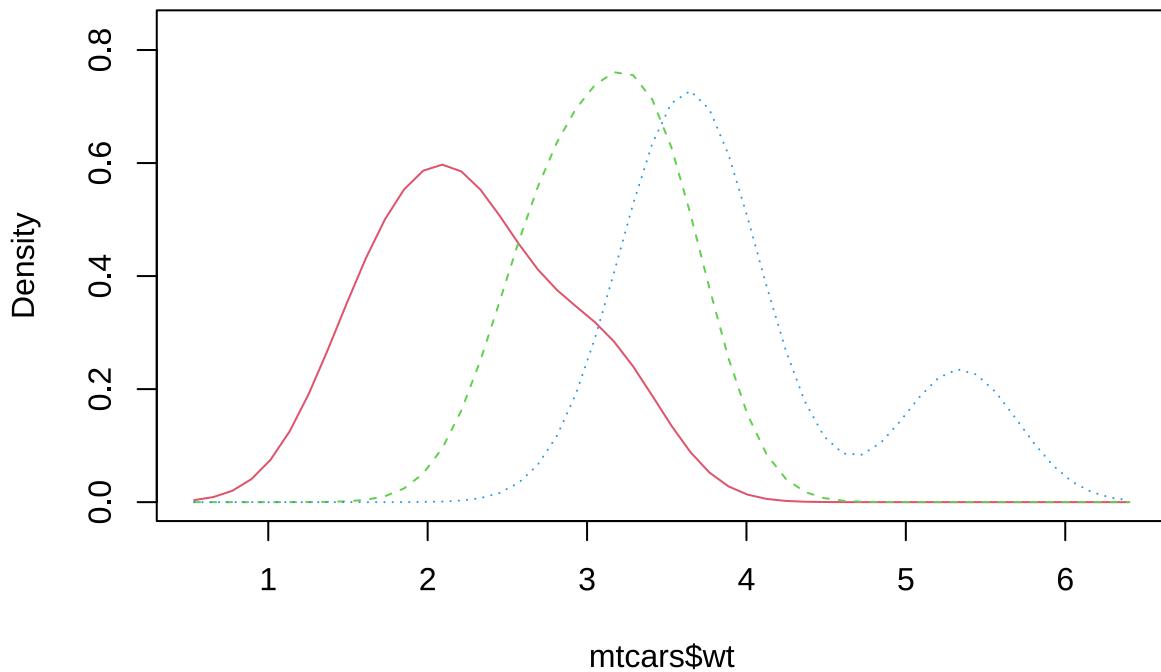
概念介绍：sm 包中 `sm.density.compare` 函数用于绘制核密度图，核密度图如果想用一条密度曲线而不是通过柱状来展示连续型变量的分布。

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
```



### 3.1.3.2 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。

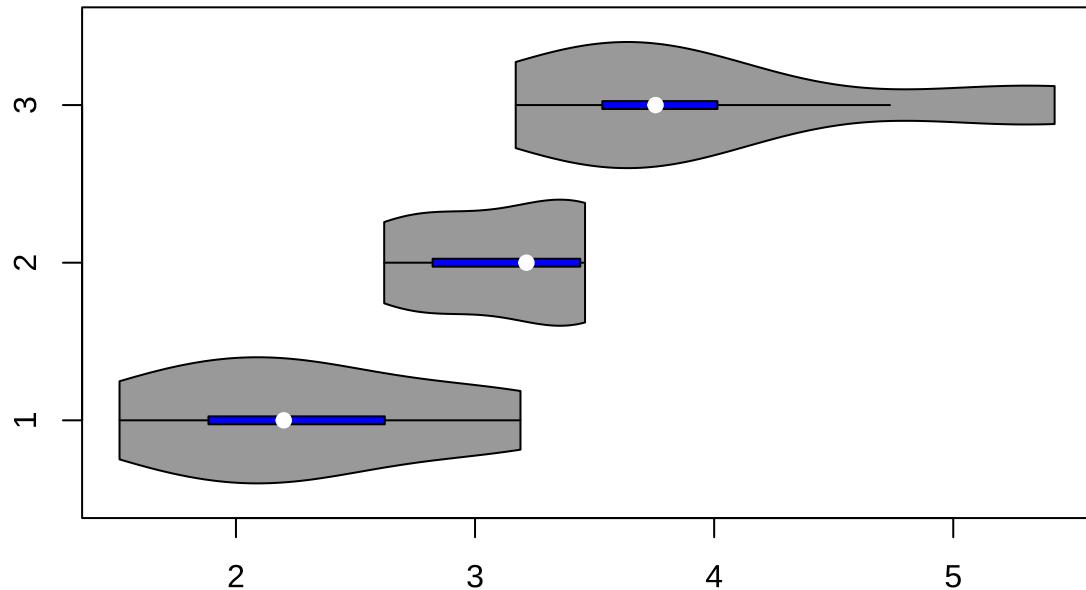
使用格式如下。

```
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。

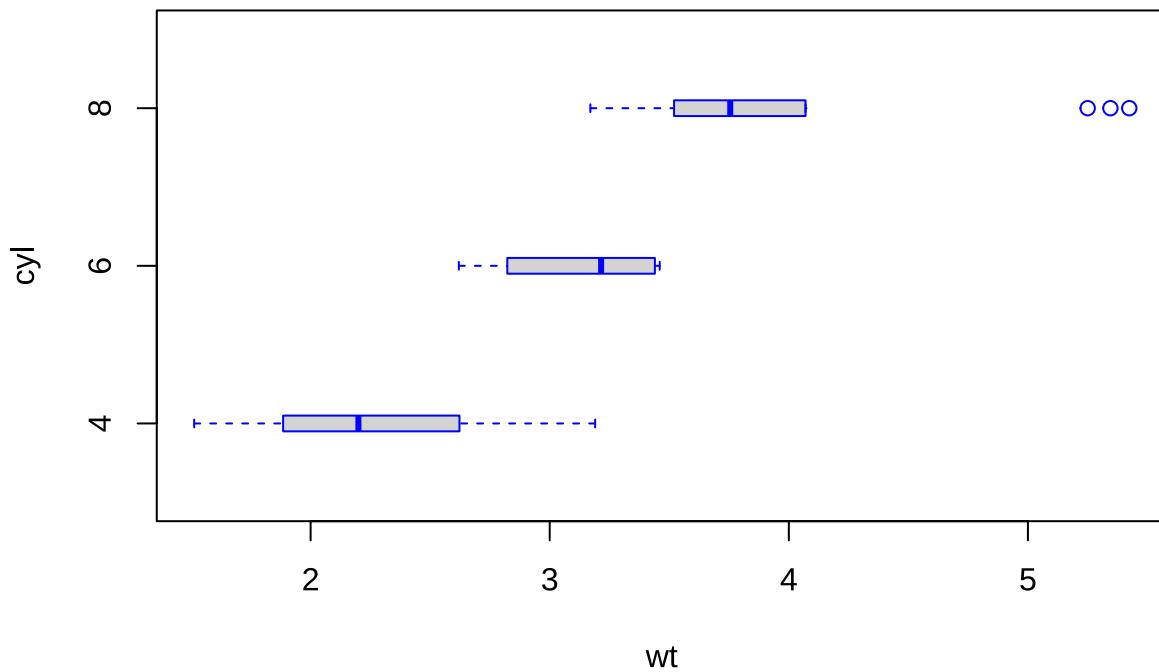
```
library(vioplot)      # 加载 vioplot 包
attach(mtcars)
vioplot(wt[cyl==4], wt[cyl==6], wt[cyl==8], border="black",
        col = "gray60", rectCol = "blue", horizontal = TRUE,
        main = '小提琴图') # 绘制小提琴图
```

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



### 3.1.3.3 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。

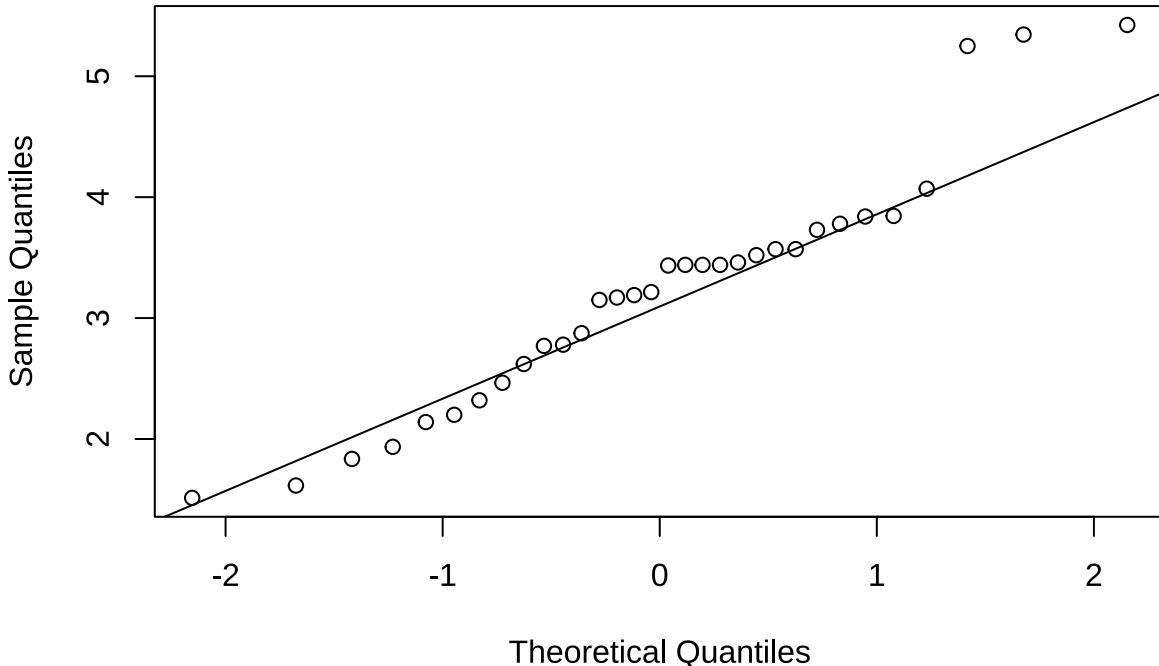
使用格式如下：

```
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**



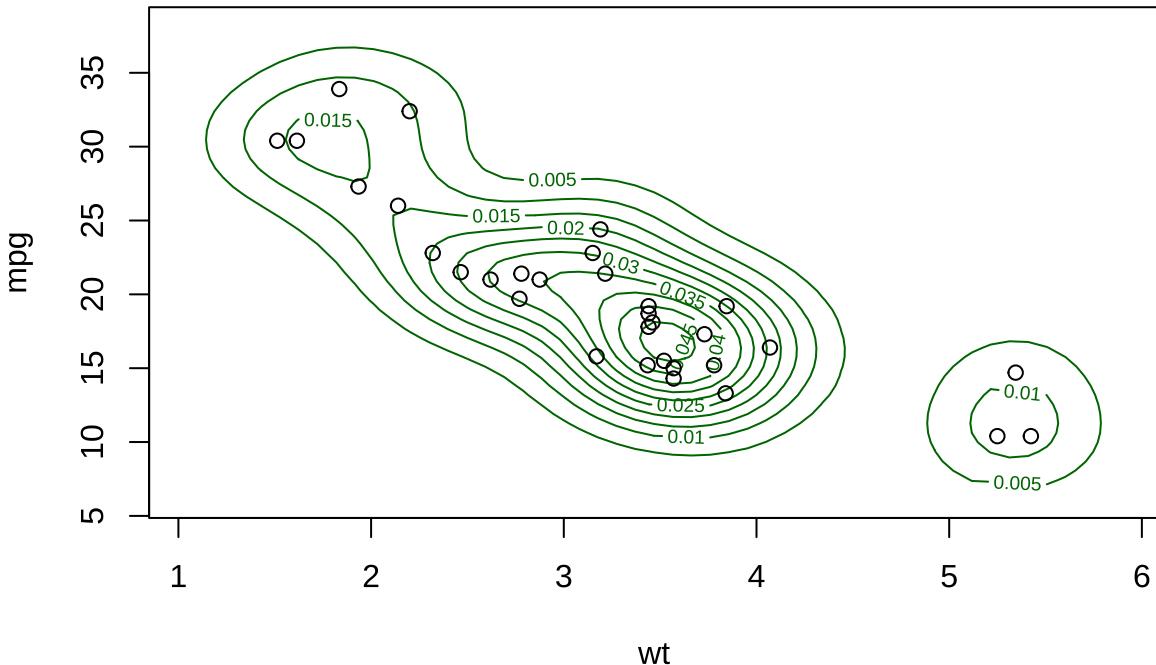
### 3.1.3.4 等高图

**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") # 画等高图
points(mtcars1) # 添加散点
```



## 3.2 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。

### 3.2.1 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：

#### 3.2.1.1 固定颜色选择函数

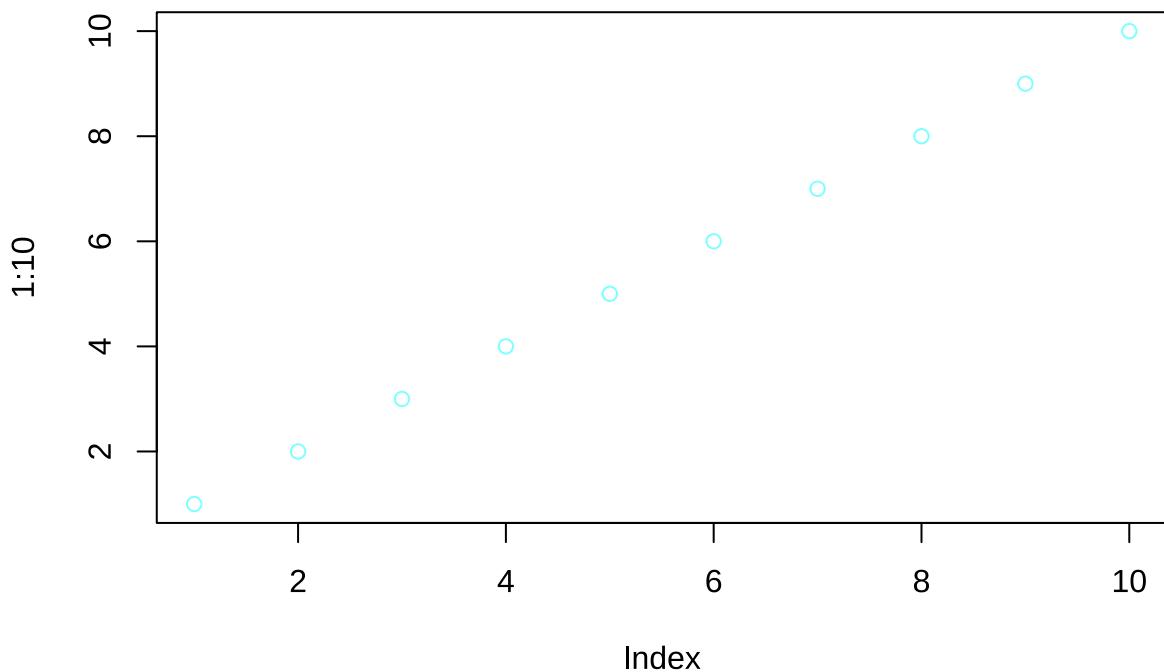
R 语言提供了自带的固定种类的颜色，主要涉及的是 `colors` 函数，该函数可以生成 657 中颜色名称，代表 657 种颜色，可以通过以下代码查看 R 自带颜色的前 20 中颜色的名称。

```
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）。

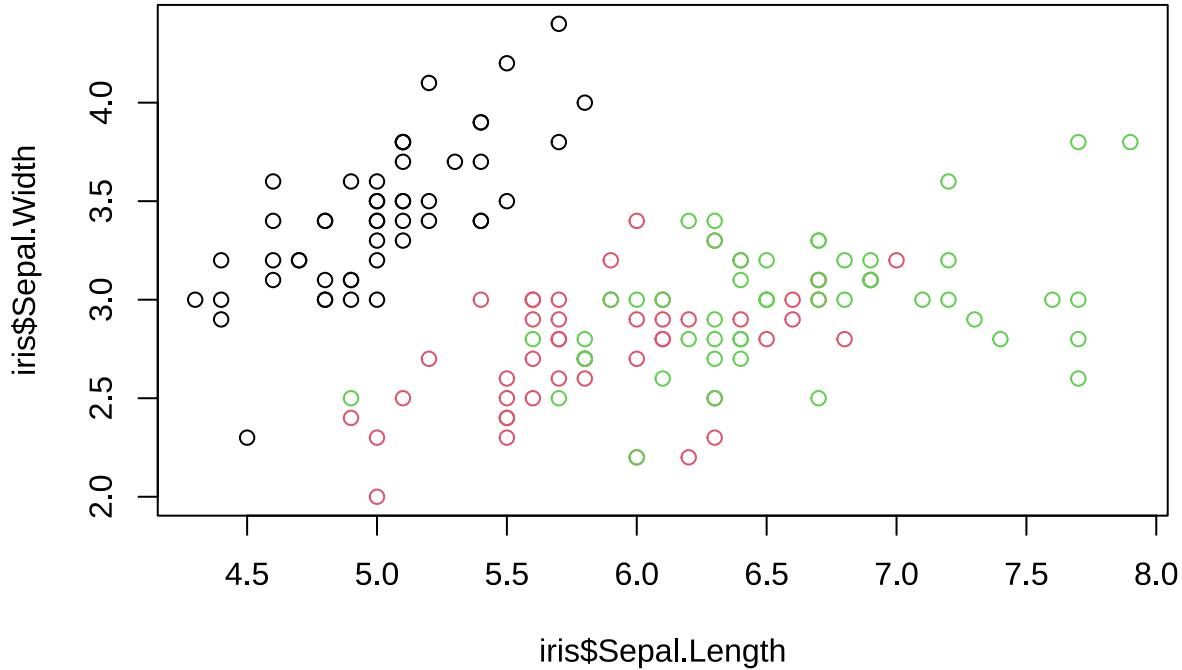
```
palette() # 返回当前的调色板设置，此时为默认值
## [1] "black"     "#DF536B"   "#61D04F"   "#2297E6"   "#28E2E5"   "#CDOBBC"   "#F5C710"
## [8] "gray62"

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)
```

### 3.2.1.2 渐变色生成函数

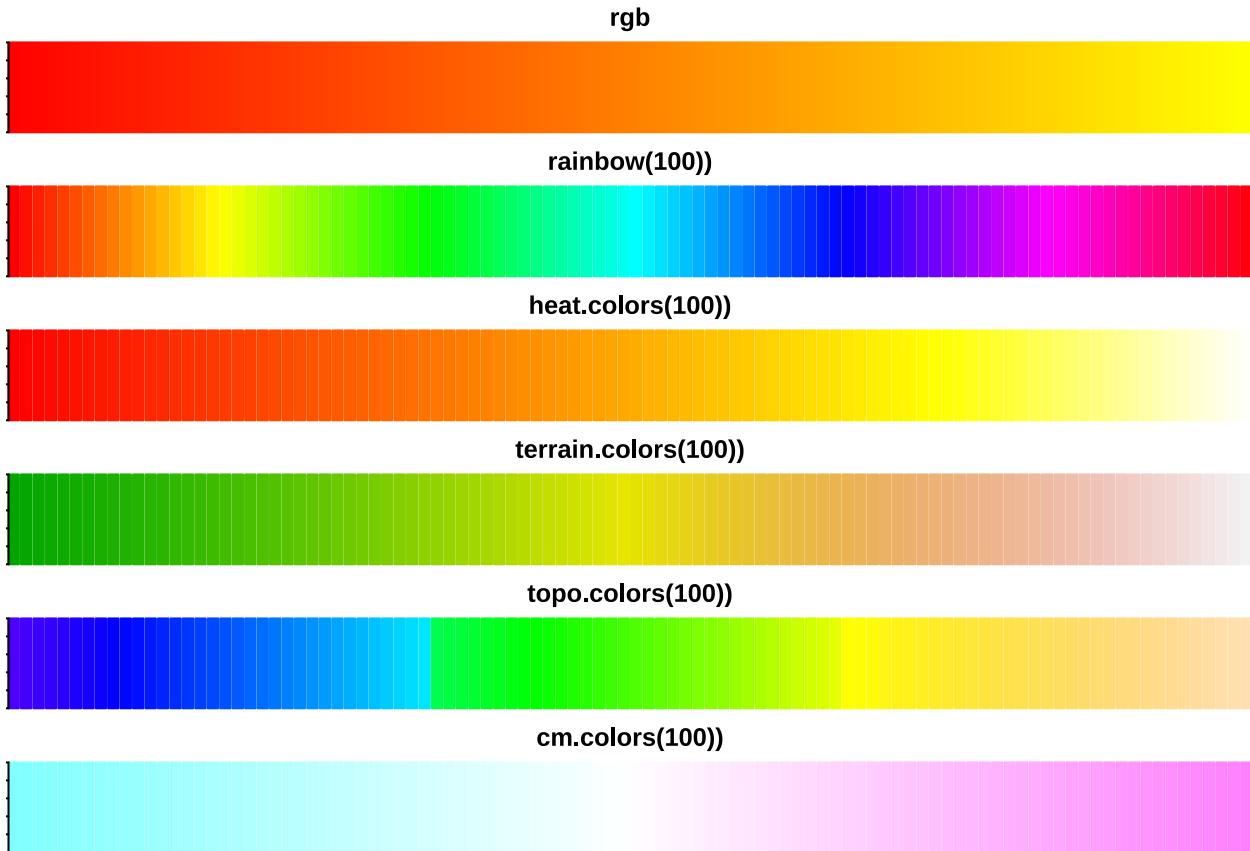
除了固定颜色选择函数外，R 还提供了一系列渐变颜色生成函数，这些函数用来控制颜色值逐步变化。

函数名称	生成原理	使用格式
<b>rgb</b>	RGB模型(红绿蓝混合)	rgb(red,green,blue,alpha, names=NULL,max=1)
<b>rainbow</b>	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)
<b>heat.colors</b>	高温、白热化(红黄白)	同 <b>rainbow</b> 函数
<b>terrain.colors</b>	地理地形(绿黄棕白)	同 <b>rainbow</b> 函数
<b>topo.colors</b>	蓝青黄棕	同 <b>rainbow</b> 函数
<b>cm.colors</b>	青白粉红	同 <b>rainbow</b> 函数
<b>brewer.pal</b>	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。



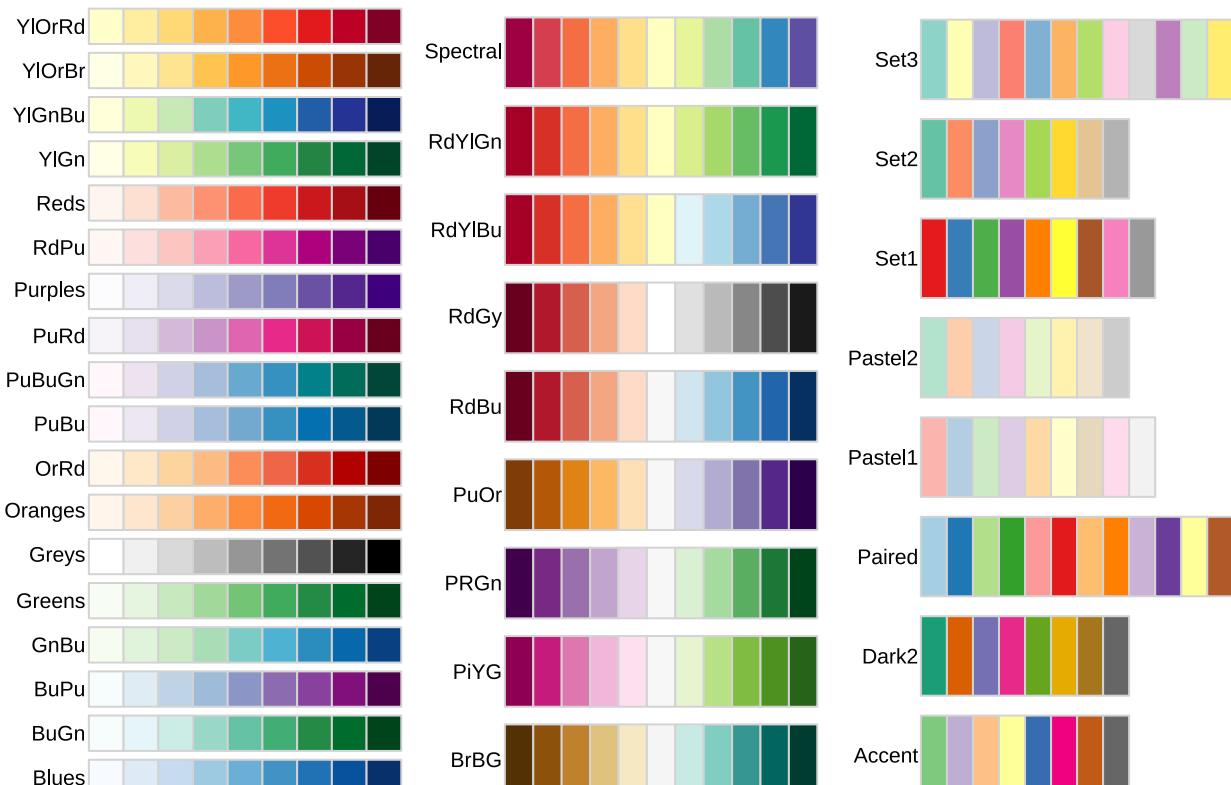
### 3.2.1.3 RColorBrewer 包

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。

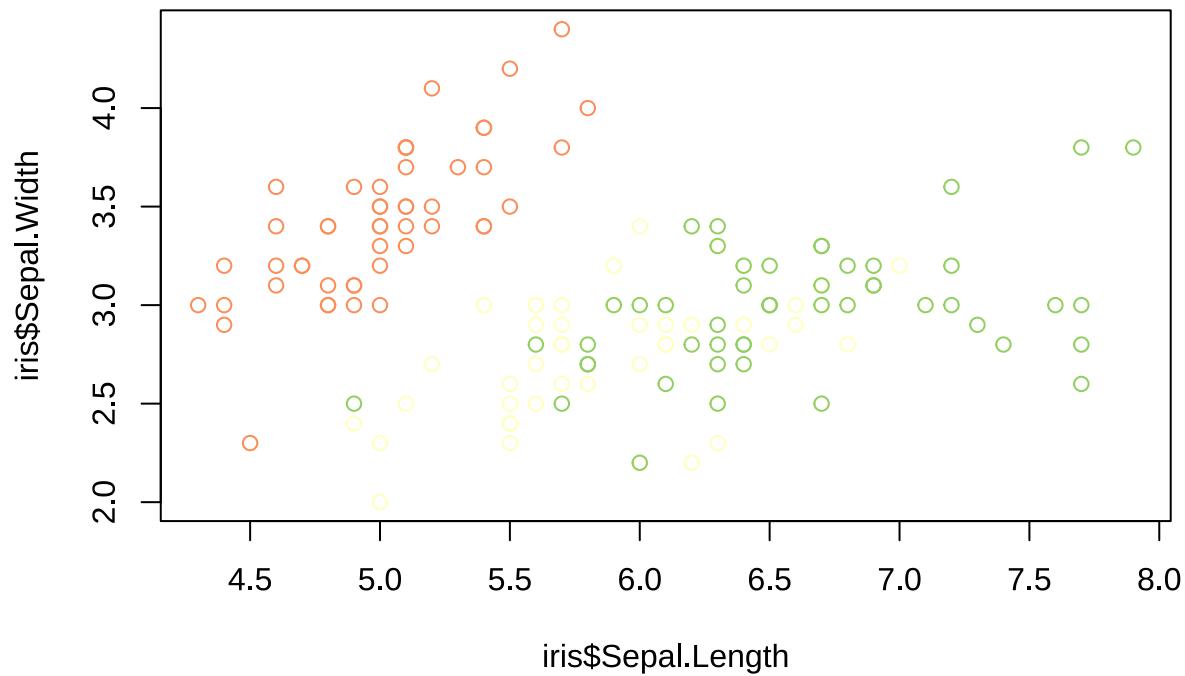
- **连续型 (Sequential)** 指生成一系列连续渐变的颜色，通常用来标记连续型数值的大小。共 18 组颜色，每组分为 9 个渐变颜色展示。
- **极端型 (Diverging)** 指生成用深色强调两端、浅色标示中部的系列颜色，可用来标记数据中的离群点。共 9 组颜色，每组分为 11 个渐变颜色展示。
- **离散型 (Qualitative)** 指生成一系列彼此差异比较明显的颜色，通常用来标记分类数据。共 8 组颜色，每组渐变颜色数不同。

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

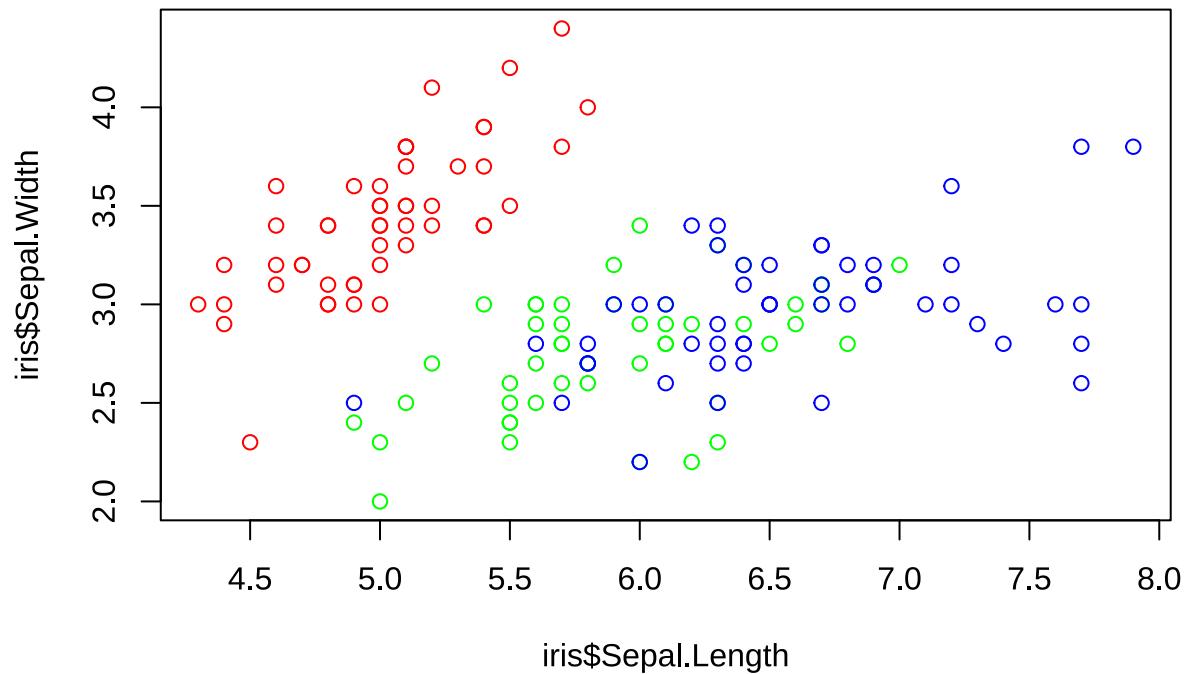
```
display.brewer.all(type="seq")
display.brewer.all(type="div")
display.brewer.all(type="qual")
```



```
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))
```



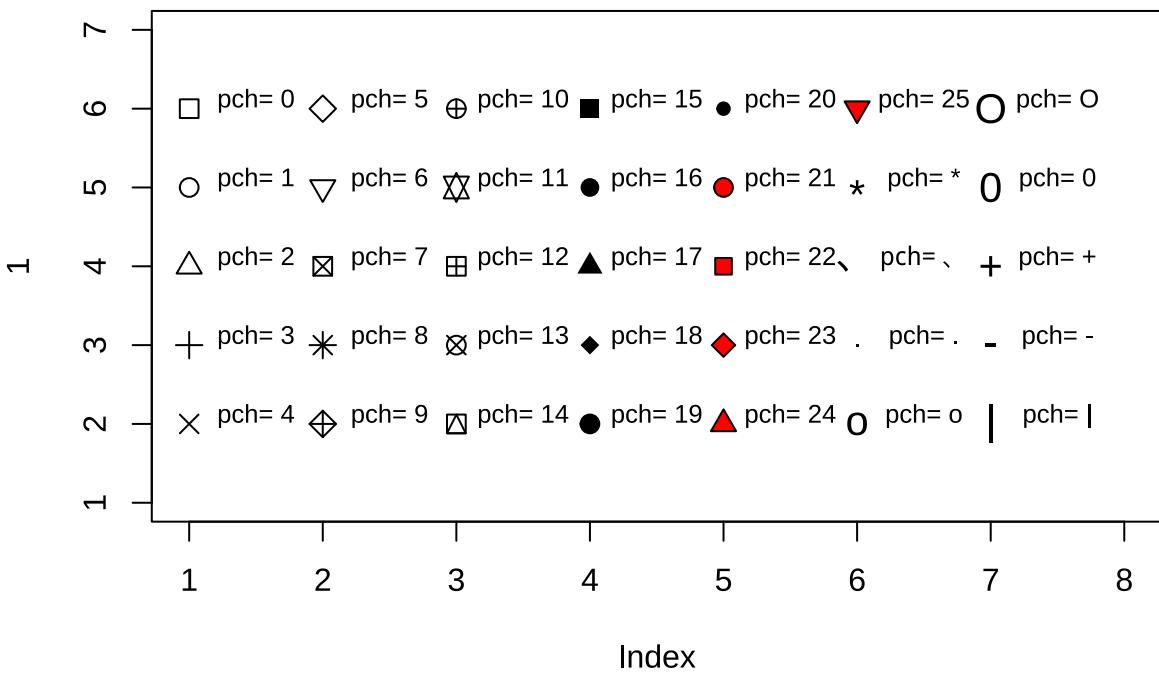
```
plot(iris$Sepal.Length, iris$Sepal.Width, col = rep(rainbow(3), each = 50))
```



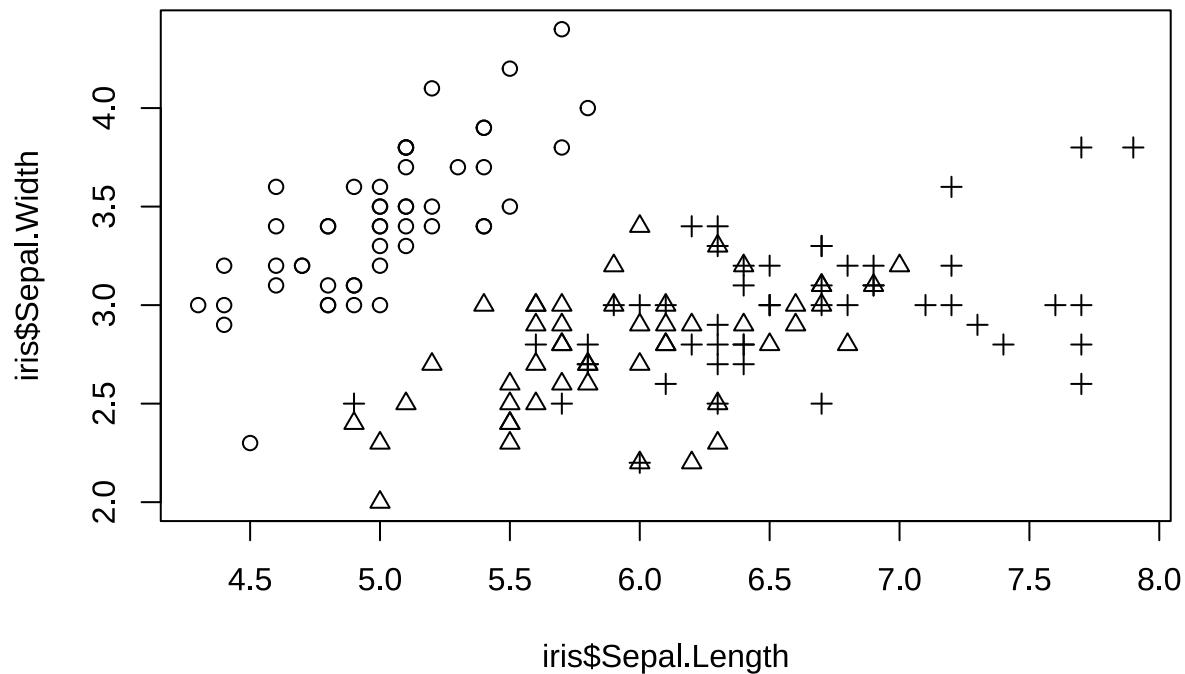
### 3.2.2 修改点符号与线条

### 3.2.2.1 点样式

参数	描述
<b>pch</b>	点的样式，取整数0-25或字符"**"," ",".".," o"," O"," 0"," +"," -","  "等
<b>cex</b>	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
<b>col</b>	点边框填充的颜色
<b>bg</b>	点内部填充的颜色，仅限21-25样式的点
<b>font</b>	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
<b>lwd</b>	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)
```

### 3.2.2.2 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

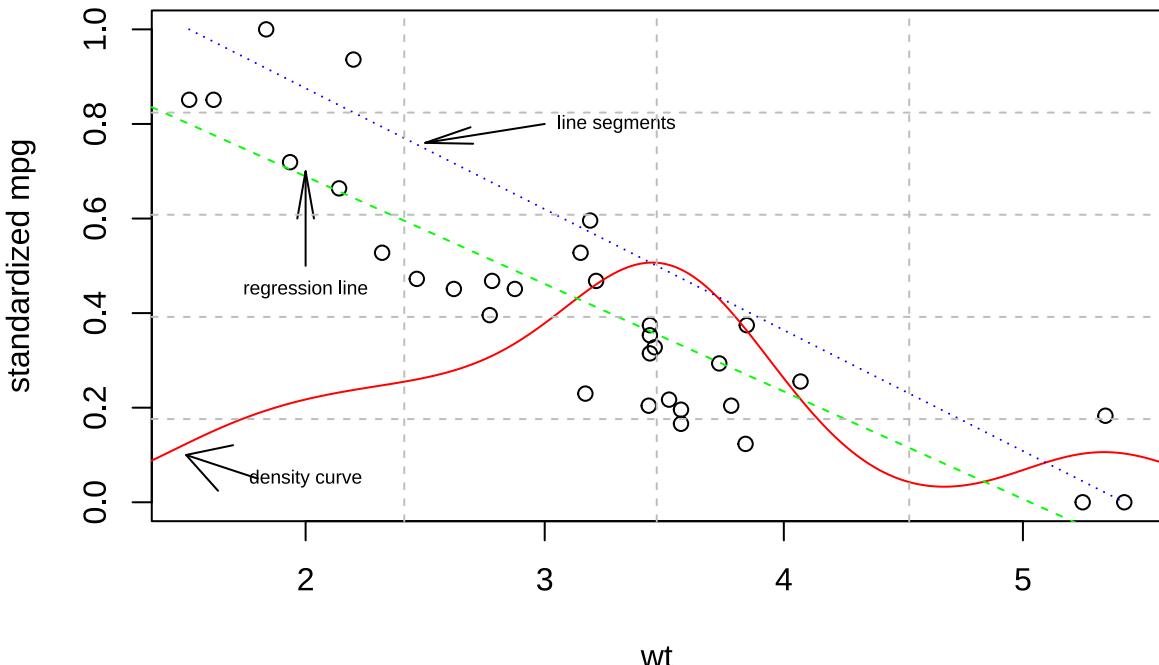
参数	描述
<b>lty</b>	线条样式, <b>0</b> 表示不画线, <b>1</b> 表示实线, <b>2</b> 表示虚线, <b>3</b> 表示点线。
<b>lwd</b>	线条粗细, <b>1</b> (默认) 表示正常宽度, 小于 <b>1</b> 表示缩放, 大于 <b>1</b> 表示放大

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")

```



### 3.2.2.3 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。

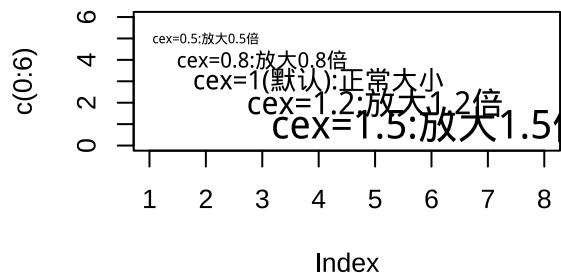
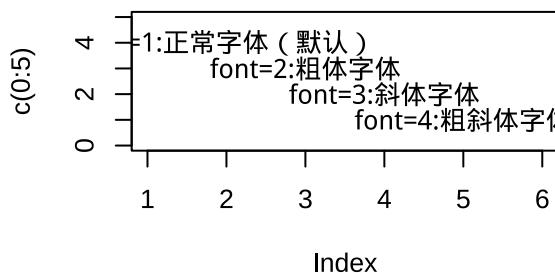
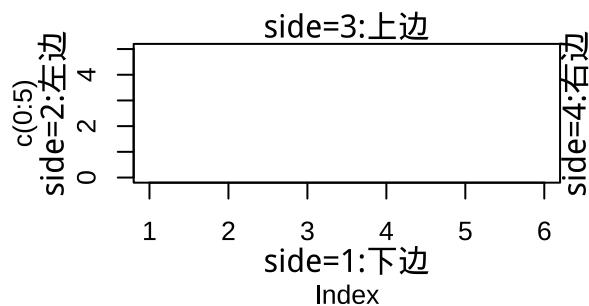
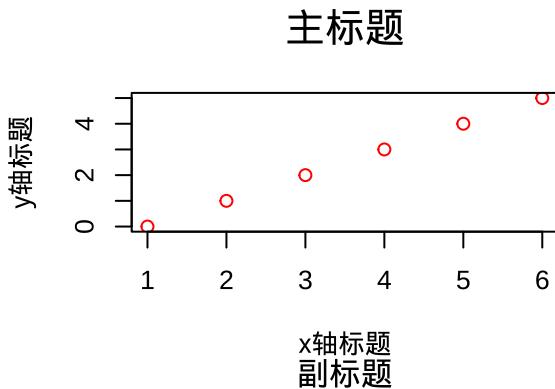
```
par(mfrow = c(2, 2))

# 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)

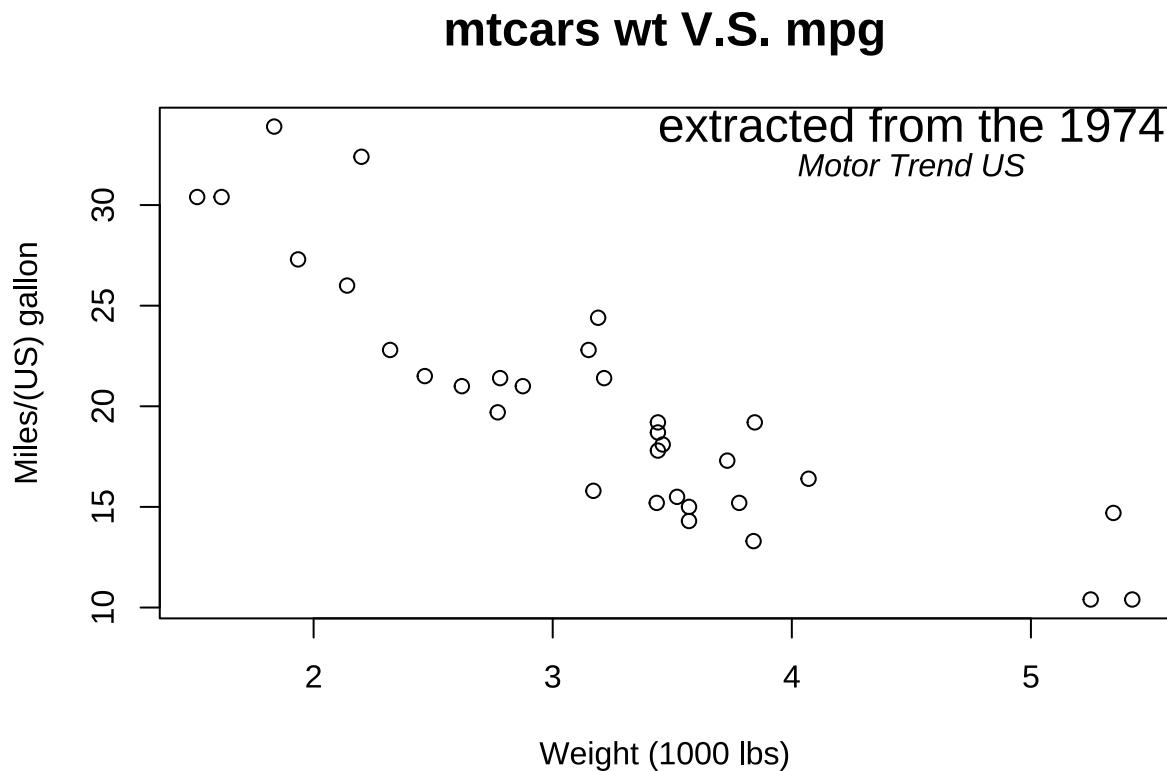
# 图三：字体展示
plot(c(0:5), col="white")
text(2,4,labels="font=1: 正常字体（默认）", font=1)
text(3,3,labels="font=2: 粗体字体", font=2)
text(4,2,labels="font=3: 斜体字体", font=3)
text(5,1,labels="font=4: 粗斜体字体", font=4)

# 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
```



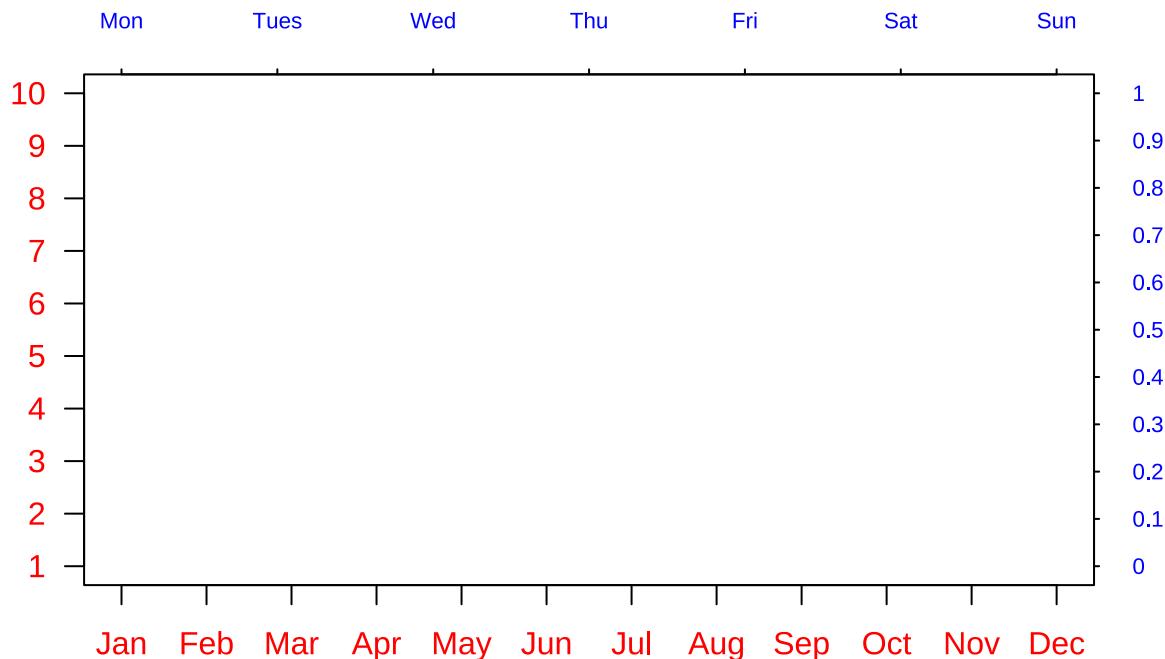
#### 3.2.2.4 设置坐标轴

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“s”，表示x /y轴以标准样式显示，取值“n”表示隐藏x /y轴
<code>xaxs/yaxs</code>	坐标轴计算方式，默认值“r”表示把原始数据的范围向外扩大4%，作为x /y轴范围，取值“l”表示x /y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x /y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)
```

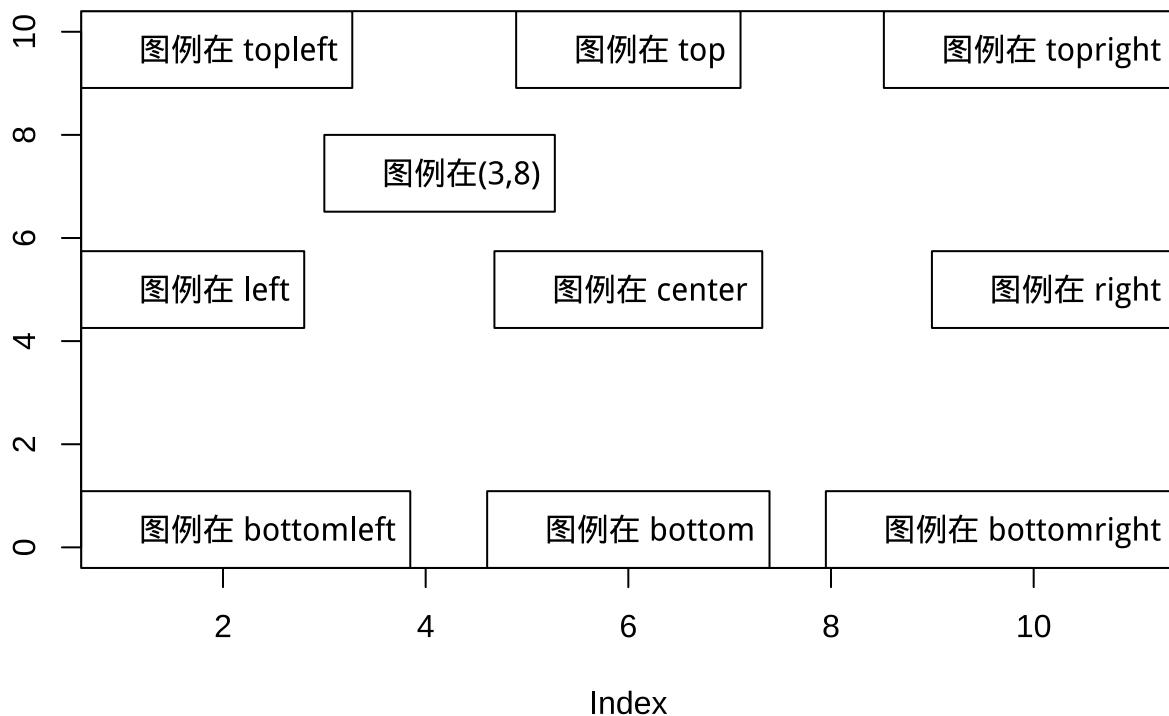
```
axis(3, at=seq(1,12,length=7), col.axis="blue", cex.axis=0.7,
     tck=-0.01, labels = c("Mon", "Tues", "Wed", "Thu", "Fri", "Sat", "Sun"))
axis(4, at=seq(1,12,length=11), col.axis="blue", cex.axis=0.7,
     tck=-0.01, labels=seq(0, 1, 0.1), las=2)
```



### 3.2.2.5 添加图例

legend 函数的绘制图例的位置效果

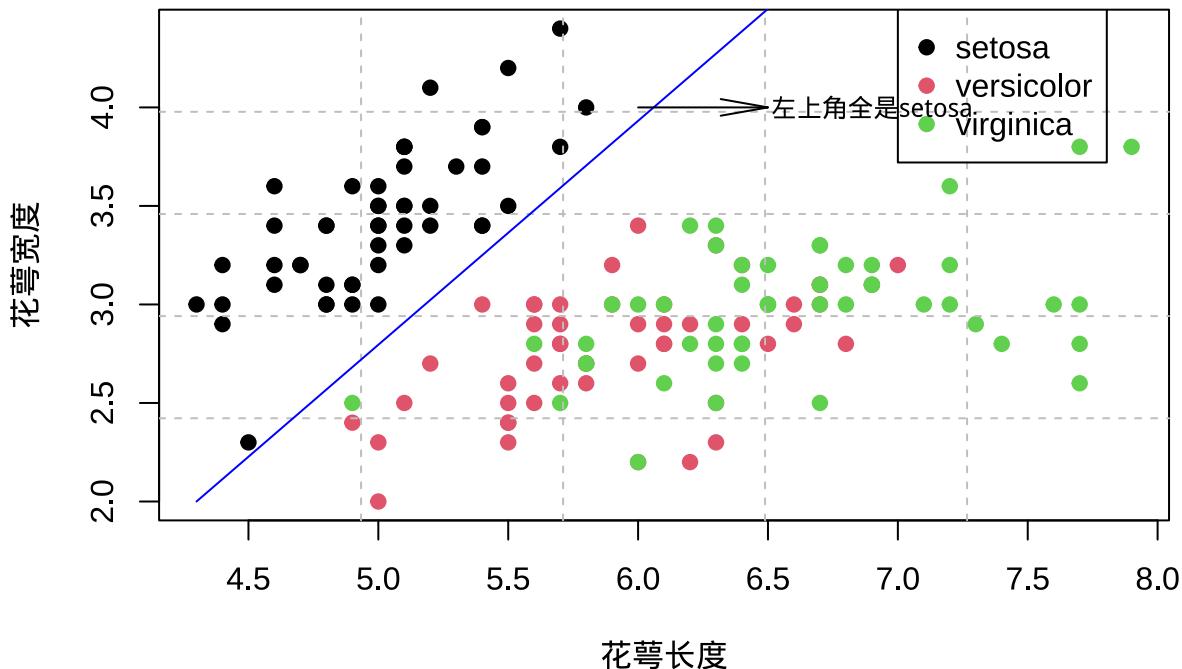
```
local=c("bottomright", "bottom", "bottomleft", "left", "topleft",
       "top", "topright", "right", "center")
par(mar = c(4,2,4,2), pty='m')
plot(c(0:10), col = "white")
legend(3, 8, " 图例在 (3,8)" )
for(i in 1:9){
  legend(local[i], paste(" 图例在", local[i]))}
```



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

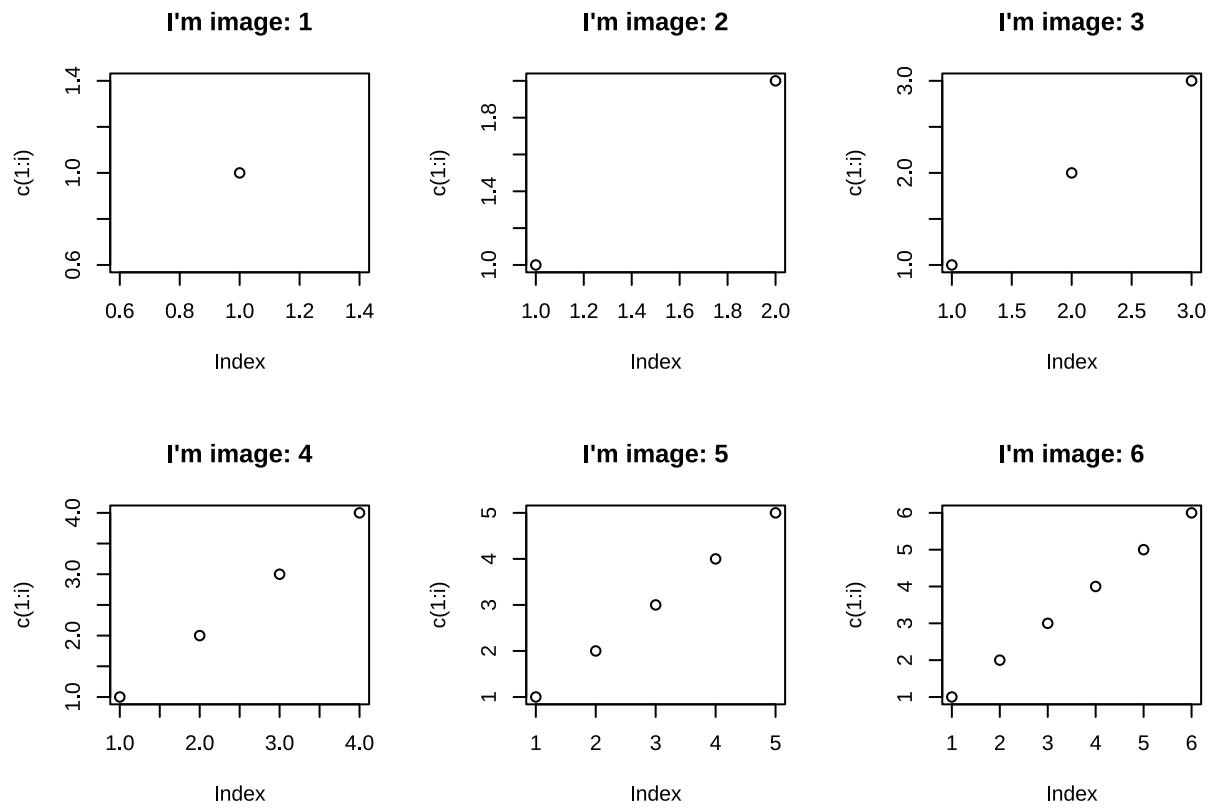


### 3.3 绘制组合图形

#### 3.3.1 par()

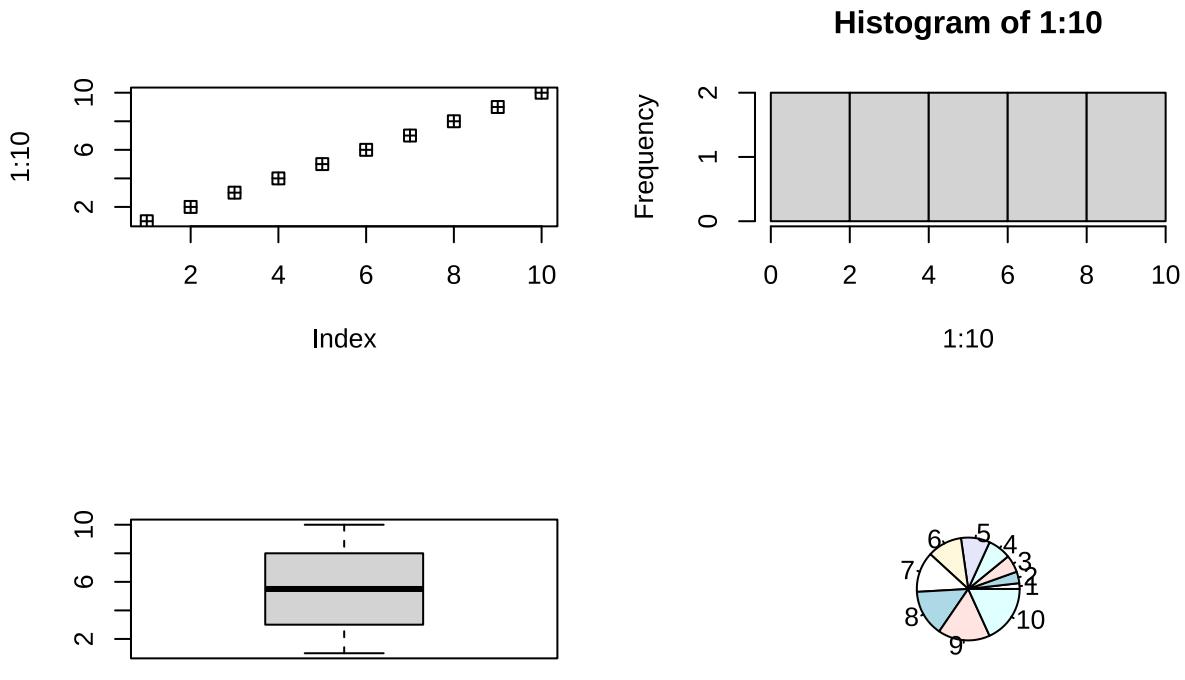
一页多图用 `mfrow` 参数或 `mfcol` 参数规定。

```
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=1)

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

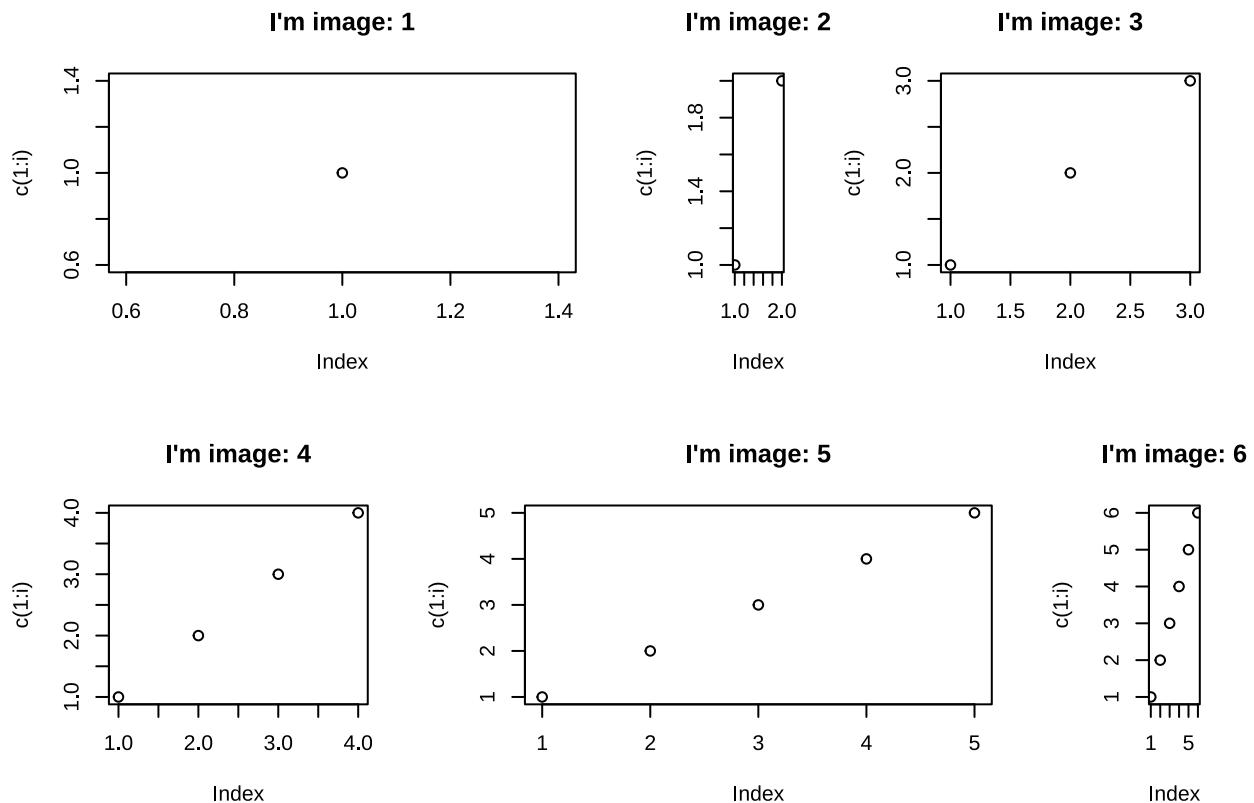


```
par(op)
```

### 3.3.2 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))
}
```



## 3.4 保存图形

### 3.4.1 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
```

### 3.4.2 在 Rstudio 窗口点击按钮保存

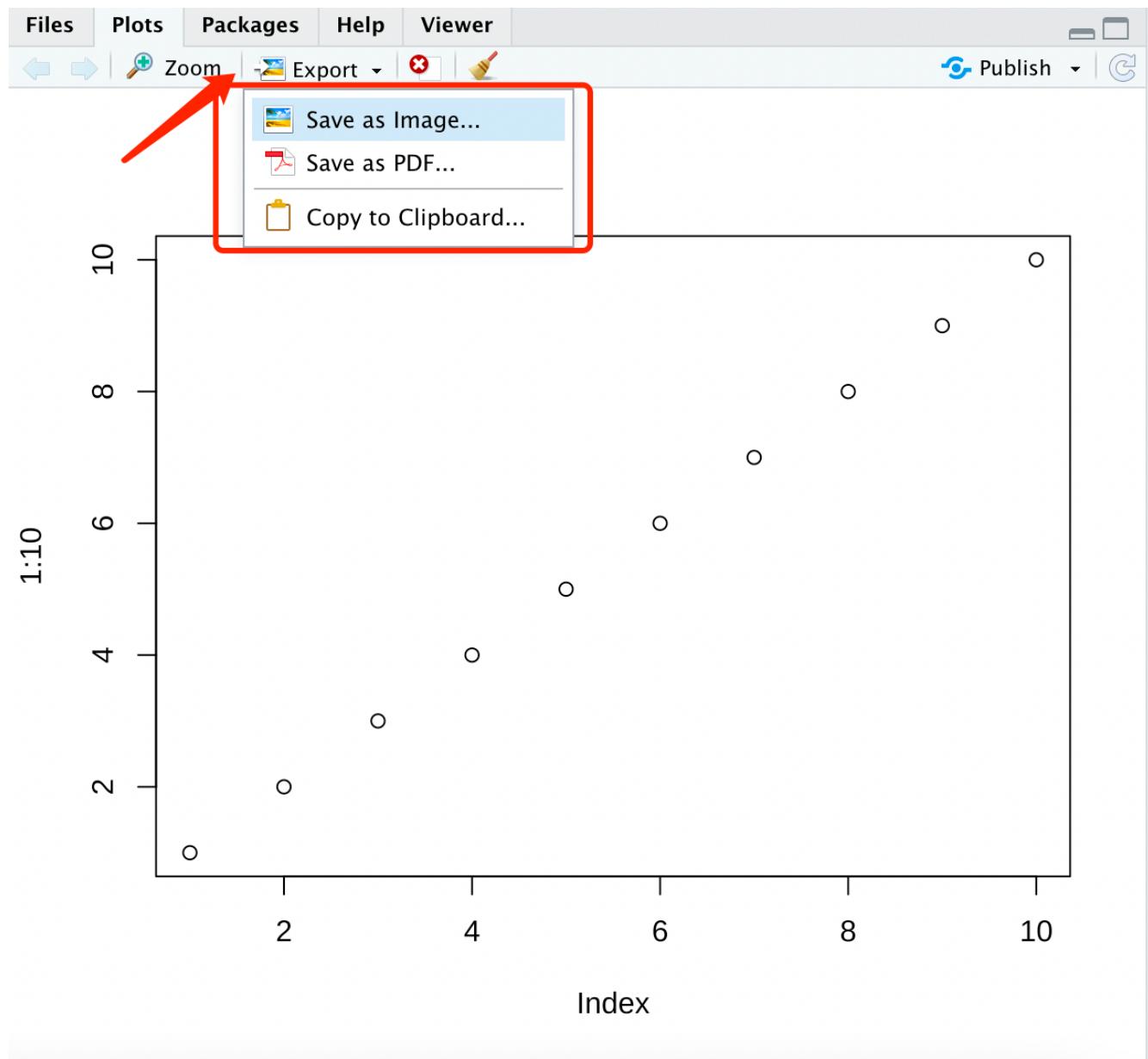


图 3.1: Rstudio 界面右下角



图 3.2: 自定义设置



# Chapter 4

## 使用 `ggplot2` 包绘图

### 4.1 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。

### 4.2 `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 4.1: 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 ...

```

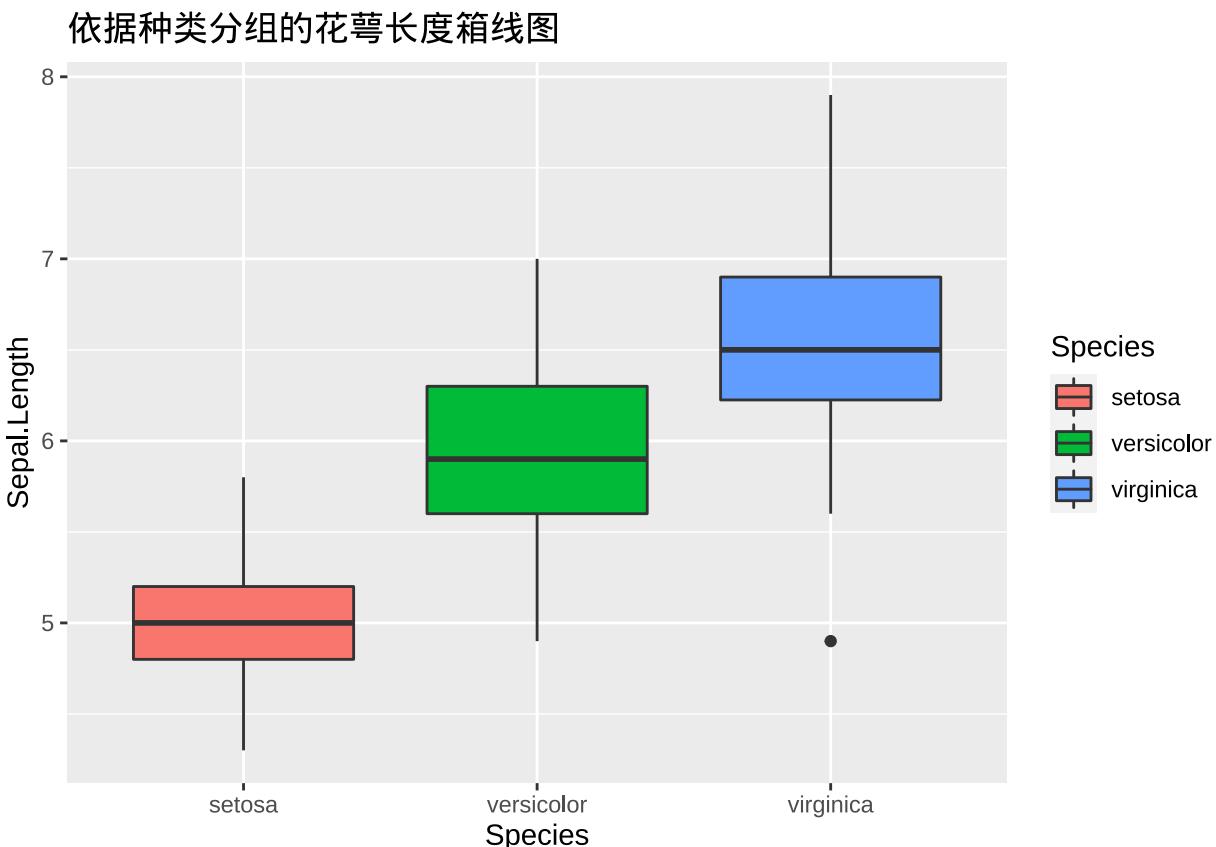
- 例子一：

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。

```

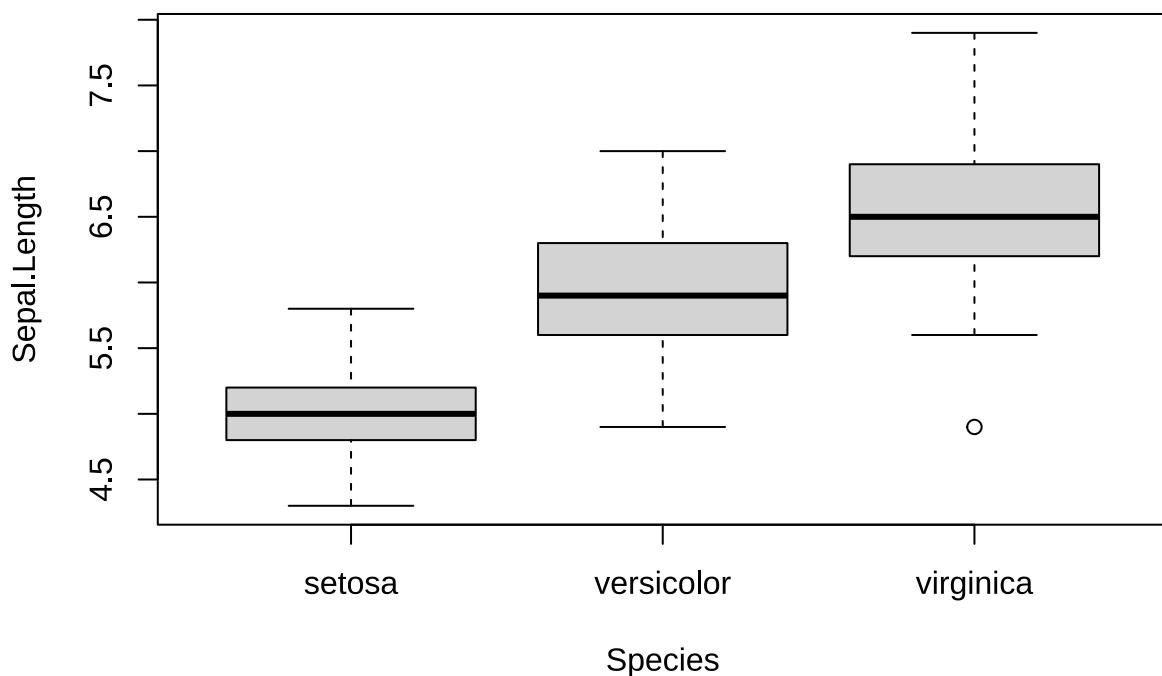
library(ggplot2)
qplot(Species, Sepal.Length, data = iris, geom = "boxplot",
      fill = Species,main = " 依据种类分组的花萼长度箱线图")

```



```
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

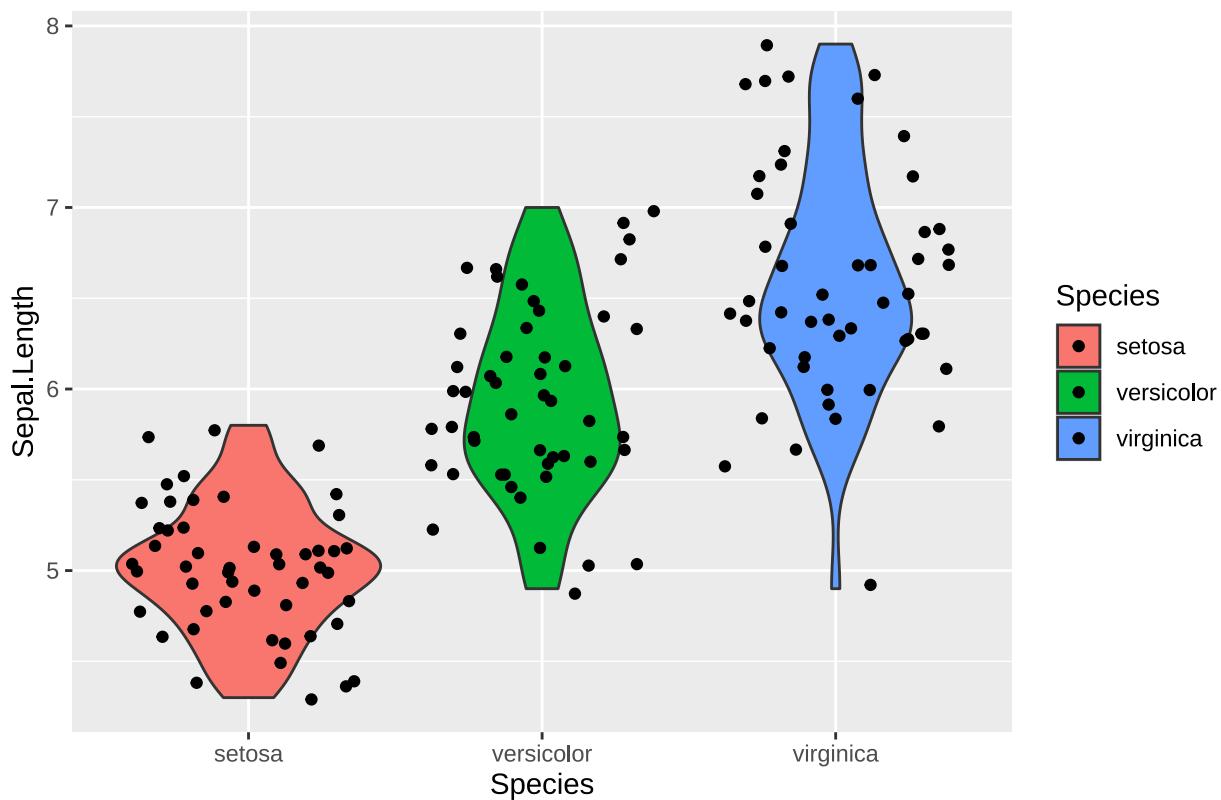


- 例子二：

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 = " 依据种类分组的花萼长度小提琴图")
```

依据种类分组的花萼长度小提琴图

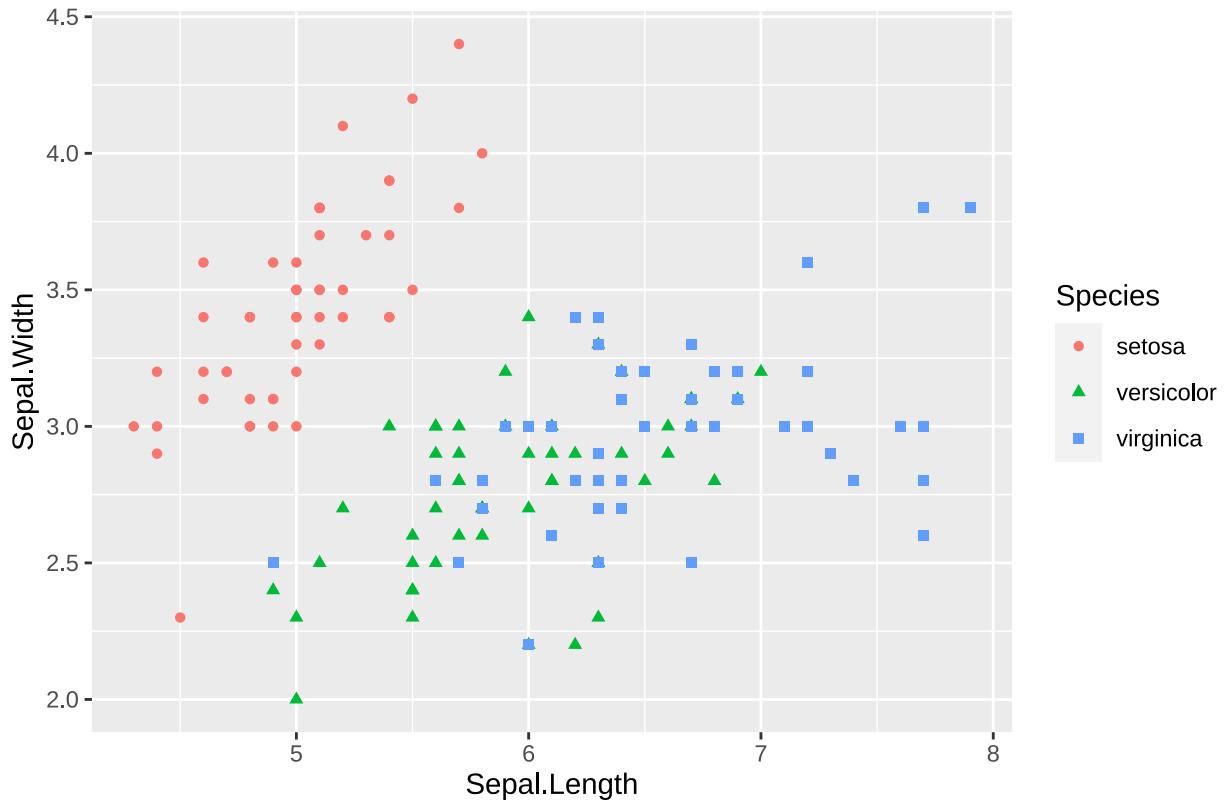


- 例子三：

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")
```

#### 绘制花萼长度和花萼宽度的散点图

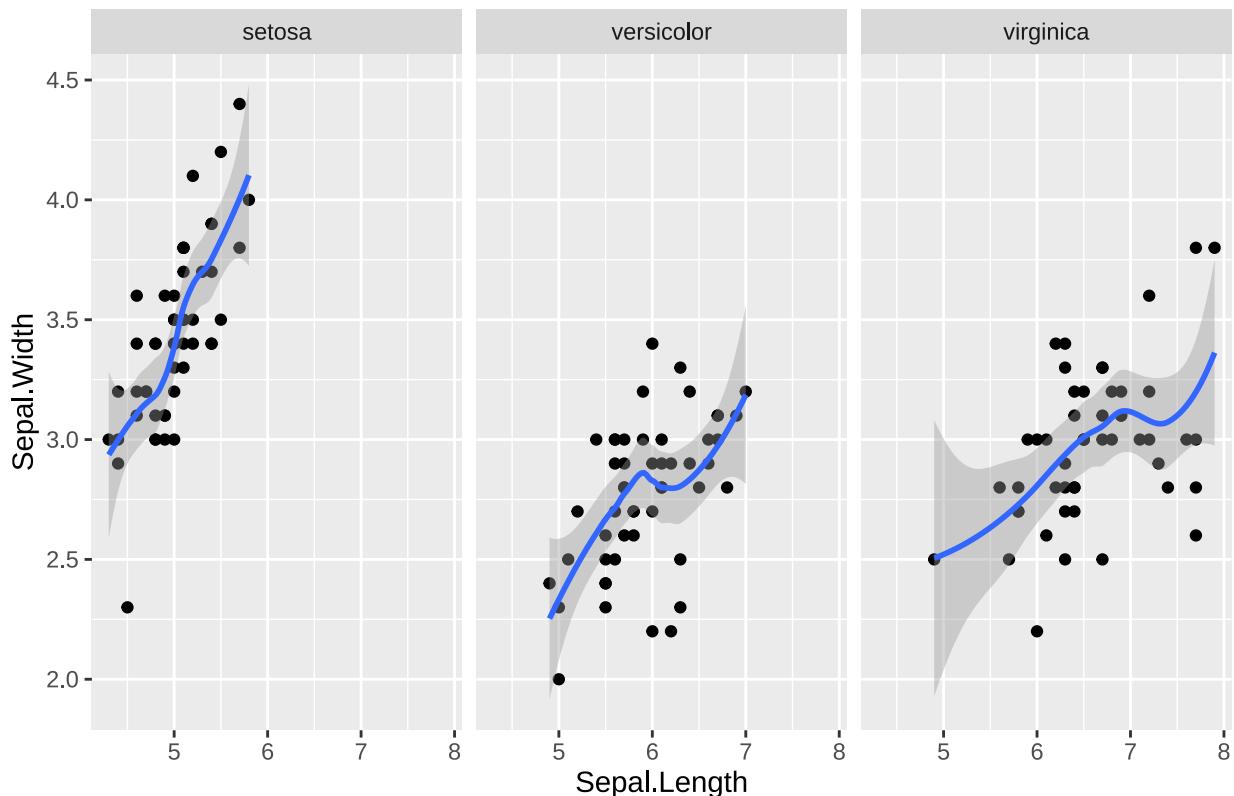


- #### • 例子四：

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



## 4.3 ggplot2 包图形语法

推荐书籍：

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

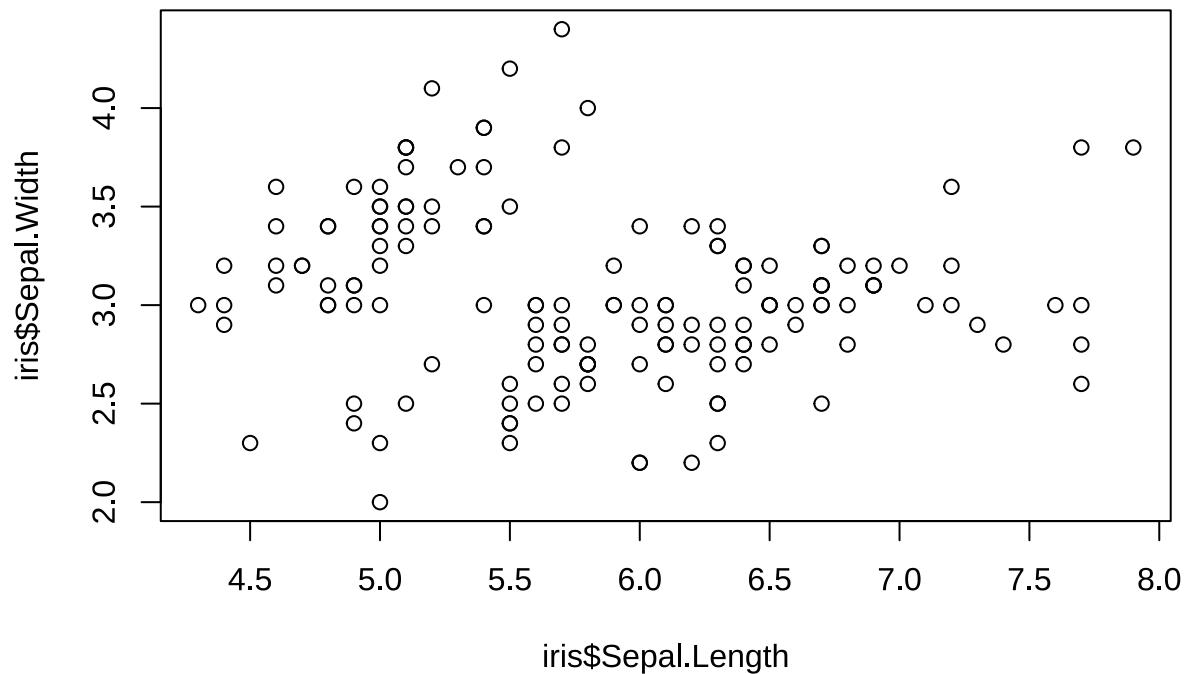
Fundamentals of Data Visualization <https://clauswilke.com/dataviz/>

### 4.3.1 对比不同画图语法

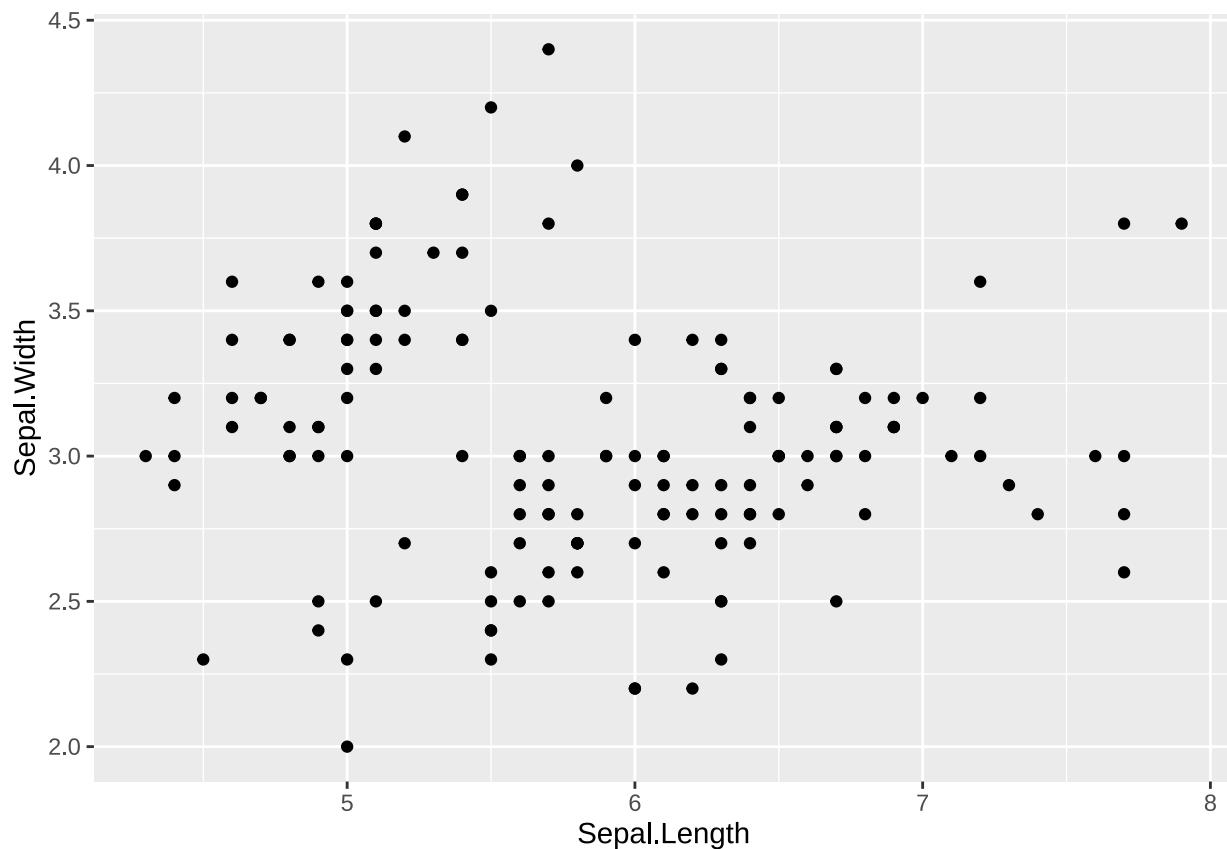
以绘制 iris 数据集中 Sepal.Length 与 Sepal.Width 的散点图为例，分别采用内置的 plot 函数与 ggplot2 包的 ggplot 函数绘制散点图，对比理解 ggplot2 包的语言逻辑。

代码（三种类型）：

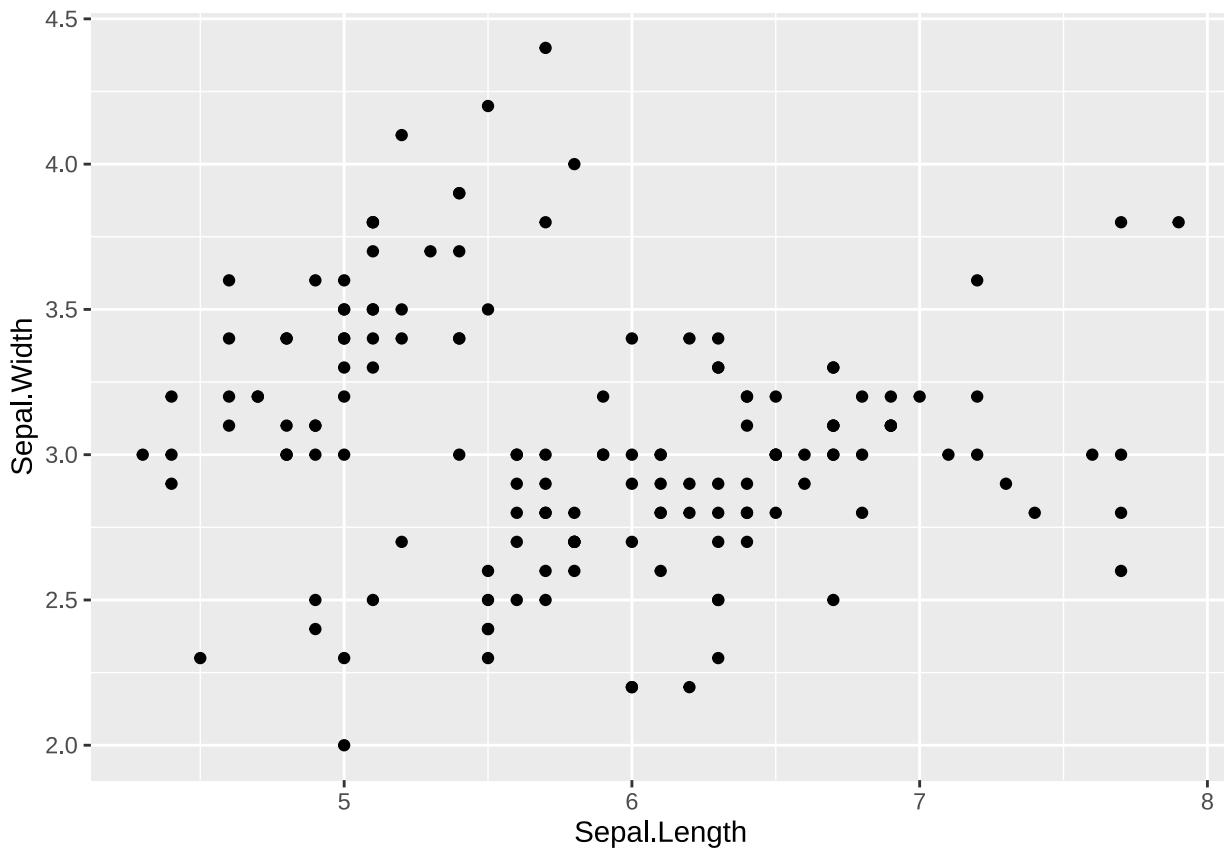
```
# 基础包
plot(iris$Sepal.Length, iris$Sepal.Width)
```



```
# 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
```



### 4.3.2 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。

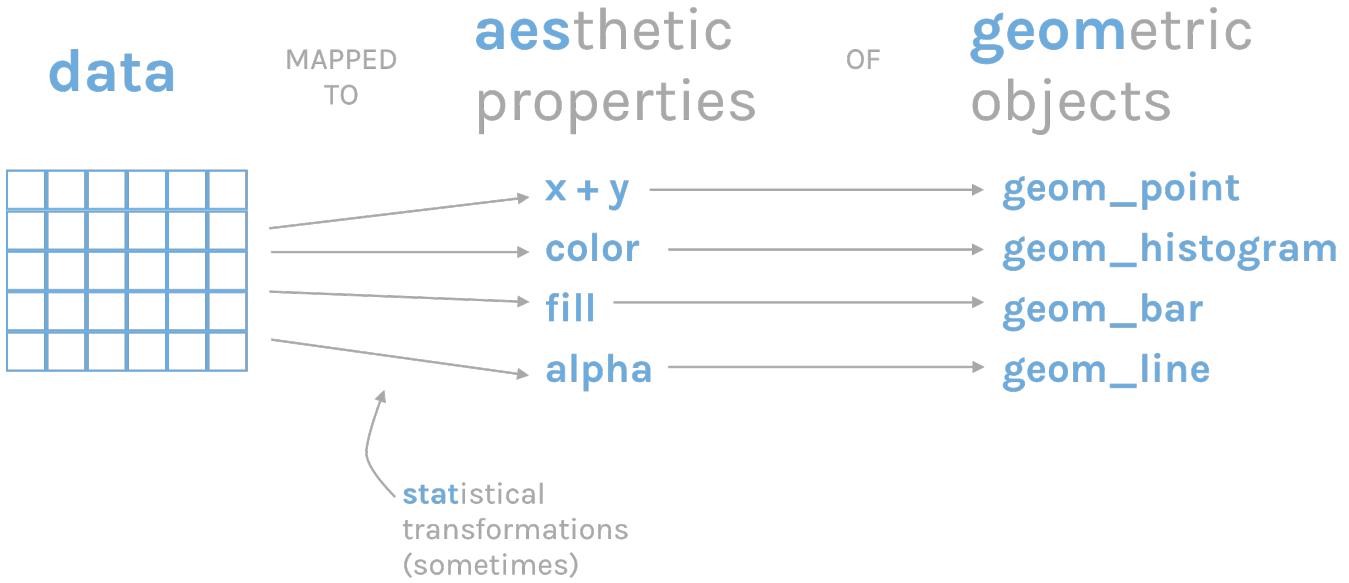
ggplot 函数包括 9 个部件：

- 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 4.2: 语法模板

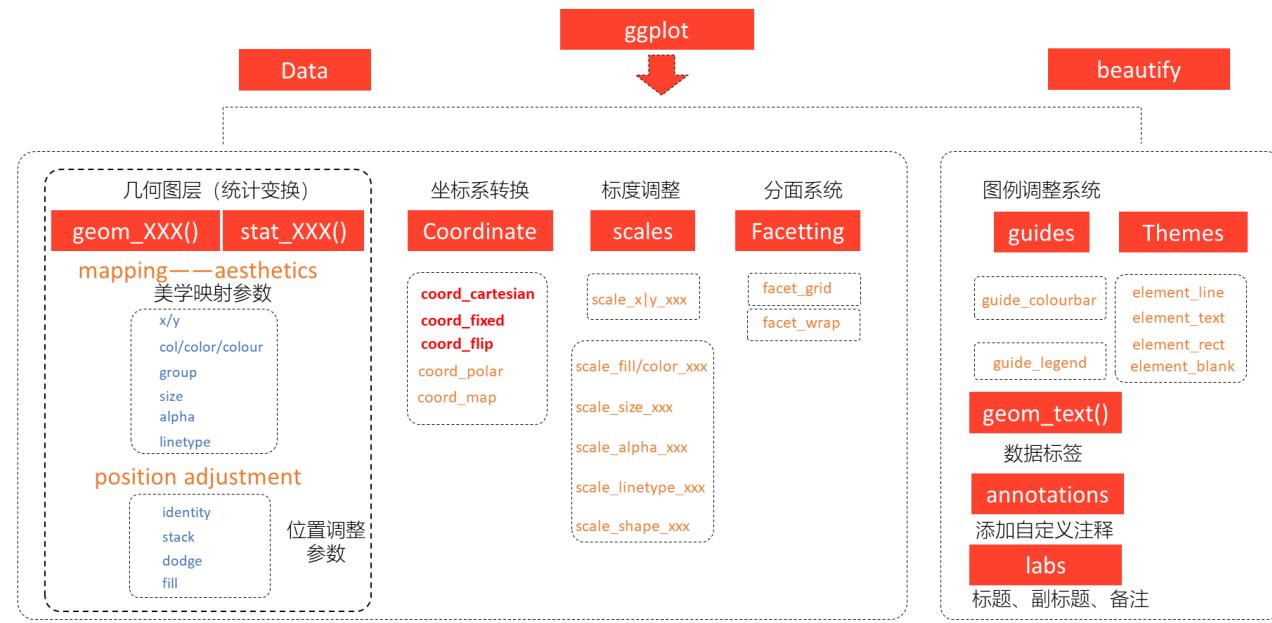
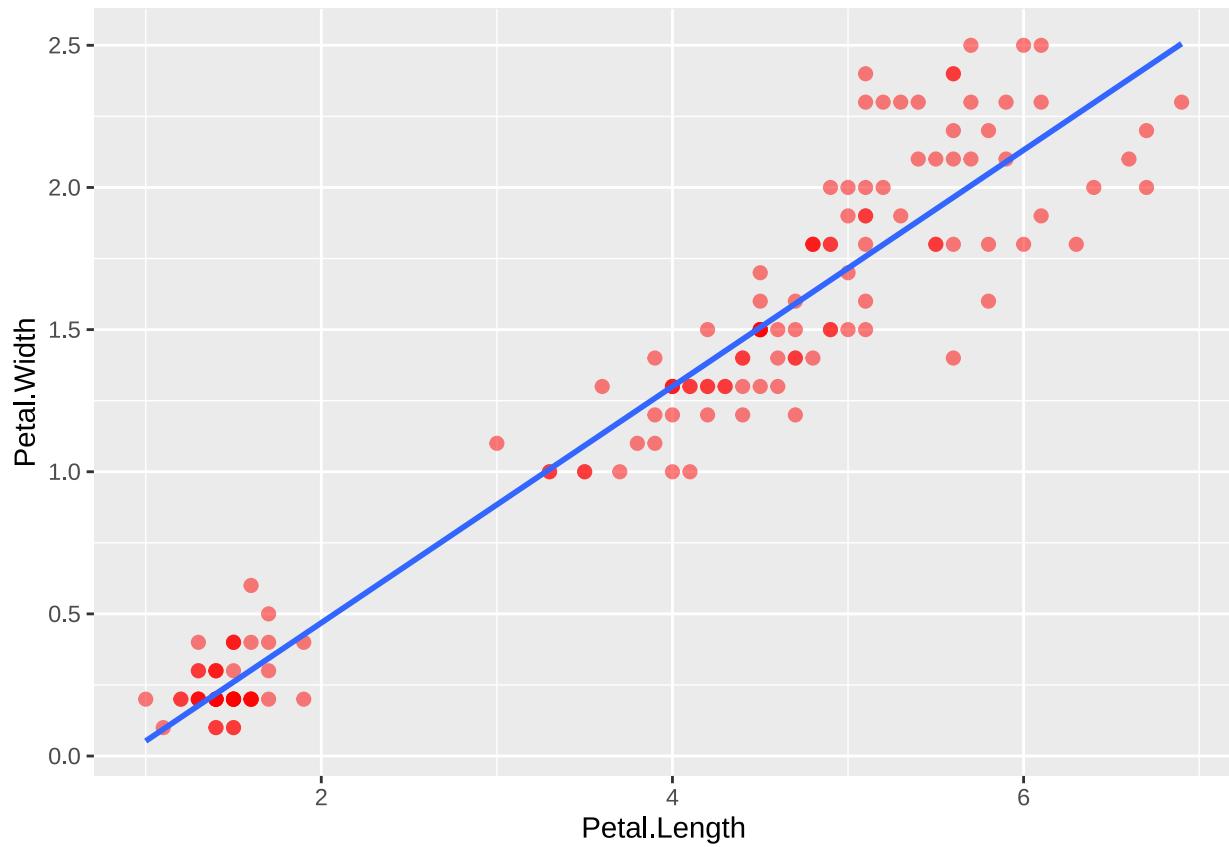
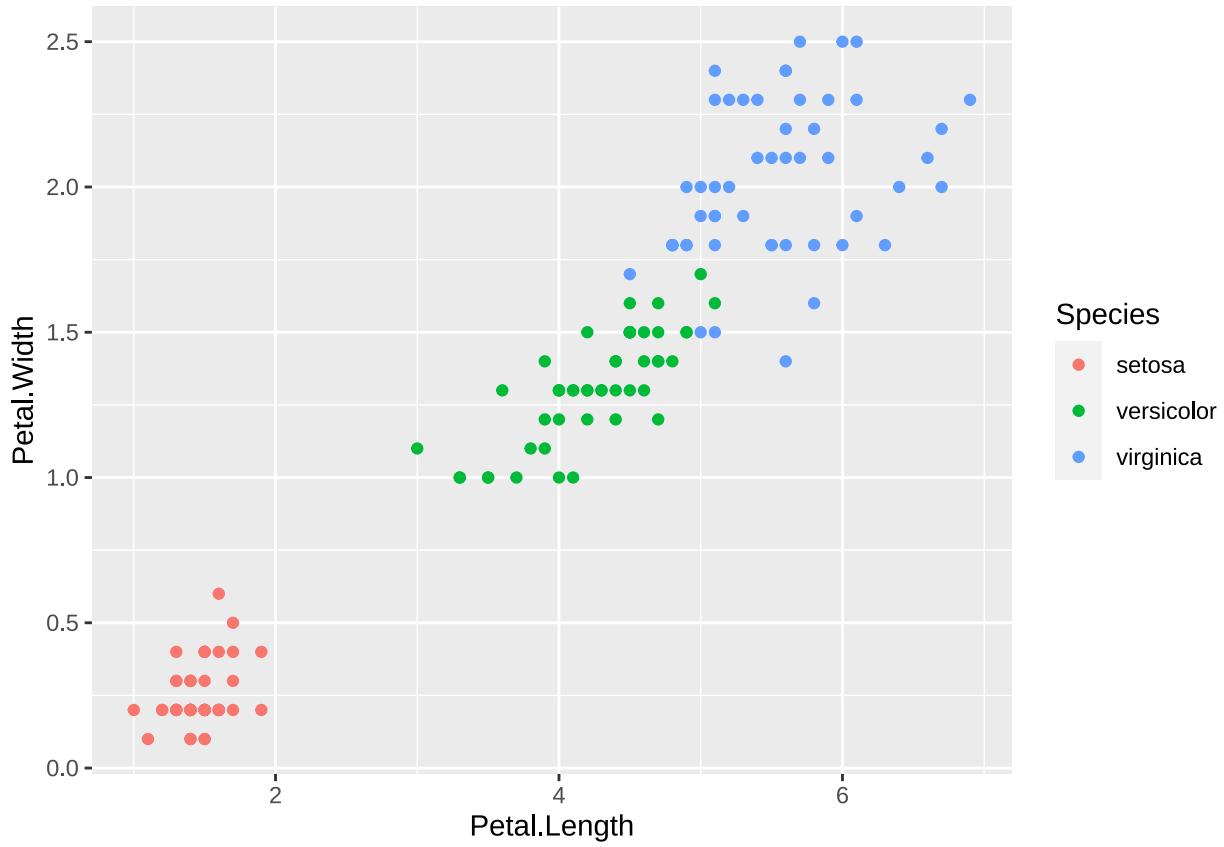


图 4.3: 把这看懂其实差不多了

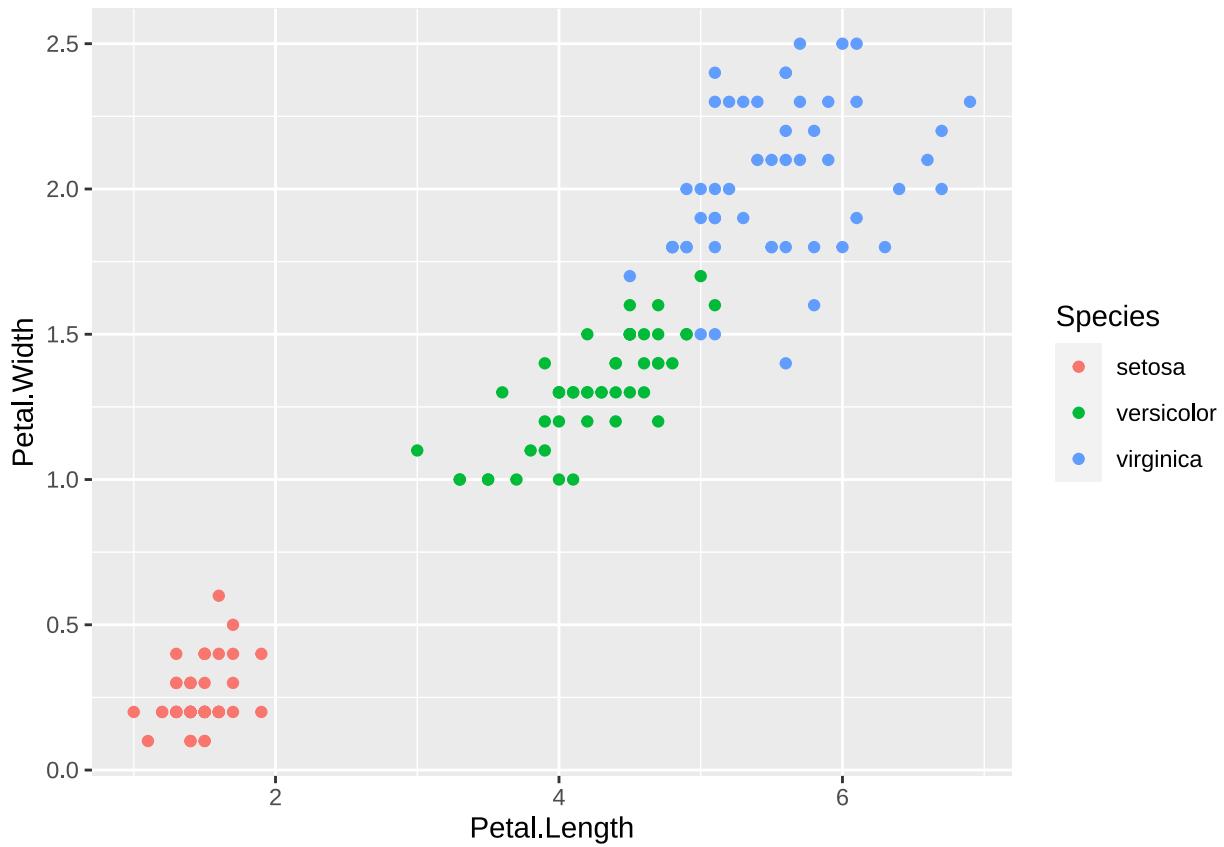


```
ggplot(data = iris, mapping = aes(x = Petal.Length, y = Petal.Width, col = Species)) +  
  geom_point()
```

#### 4.3.3 全局变量 vs. 局部变量



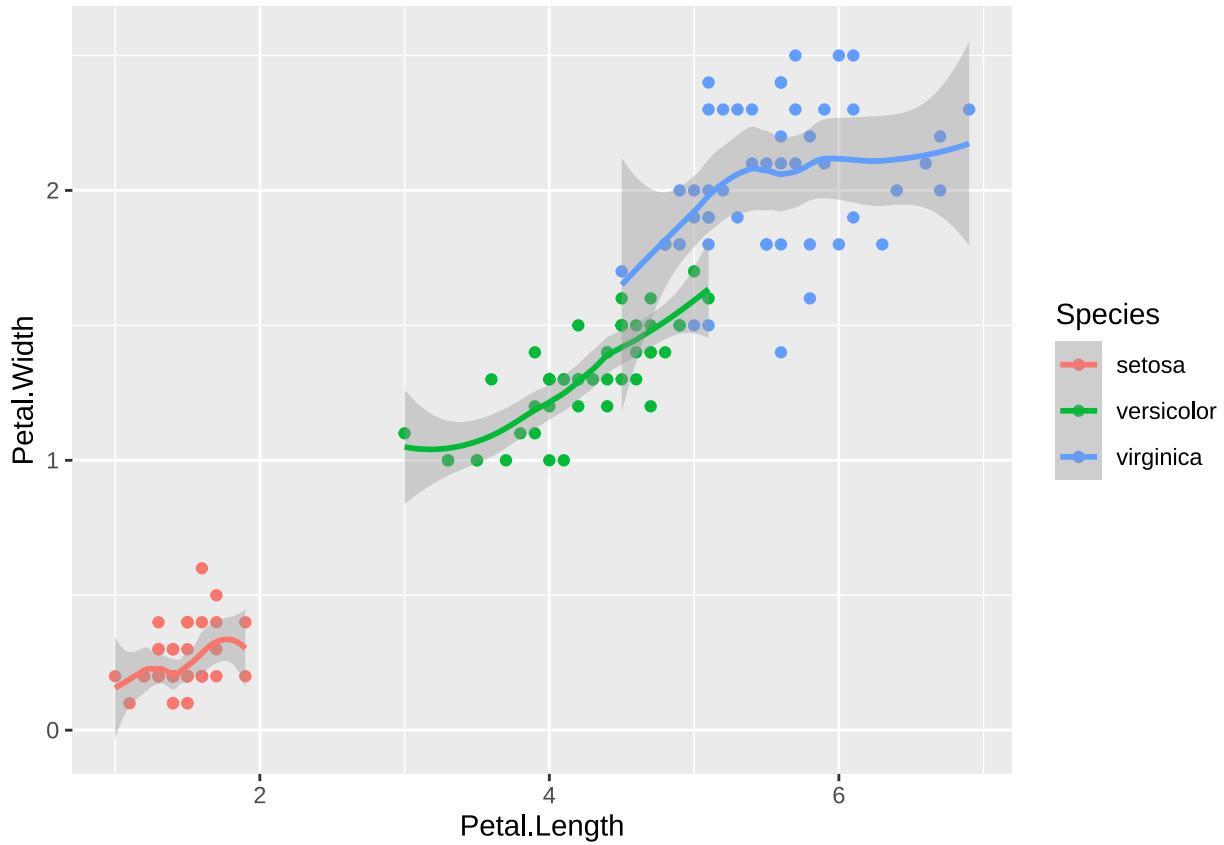
```
ggplot(data = iris) +  
  geom_point(mapping = aes(x = Petal.Length, y = Petal.Width, col = Species))
```



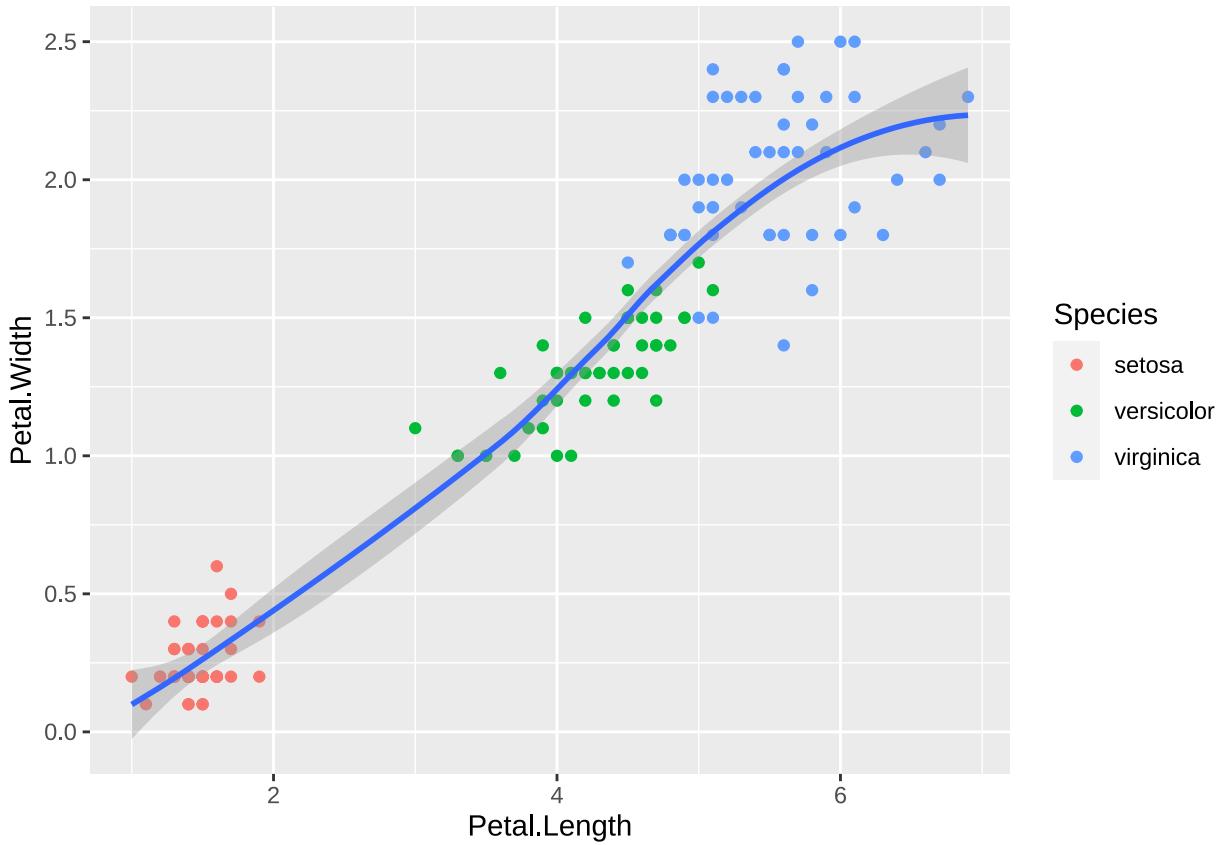
大家可以看到，以上两段代码出来的图是一样。但背后的含义却不同。

### 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(col = Species)) +
  geom_smooth()
```



## 4.4 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, ymin, ymax` 画中间有线的纵向条形;
      - \* `geom_errbar()` 对每个观测输入的 `x, ymin, ymax` 画纵向误差条;
      - \* `geom_linerange()` 对每个观测输入的 `x, ymin, ymax` 画一条竖线;
      - \* `geom_pointrnage()` 对每个观测输入的 `x, y, ymin, 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。

#### 4.4.1 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*

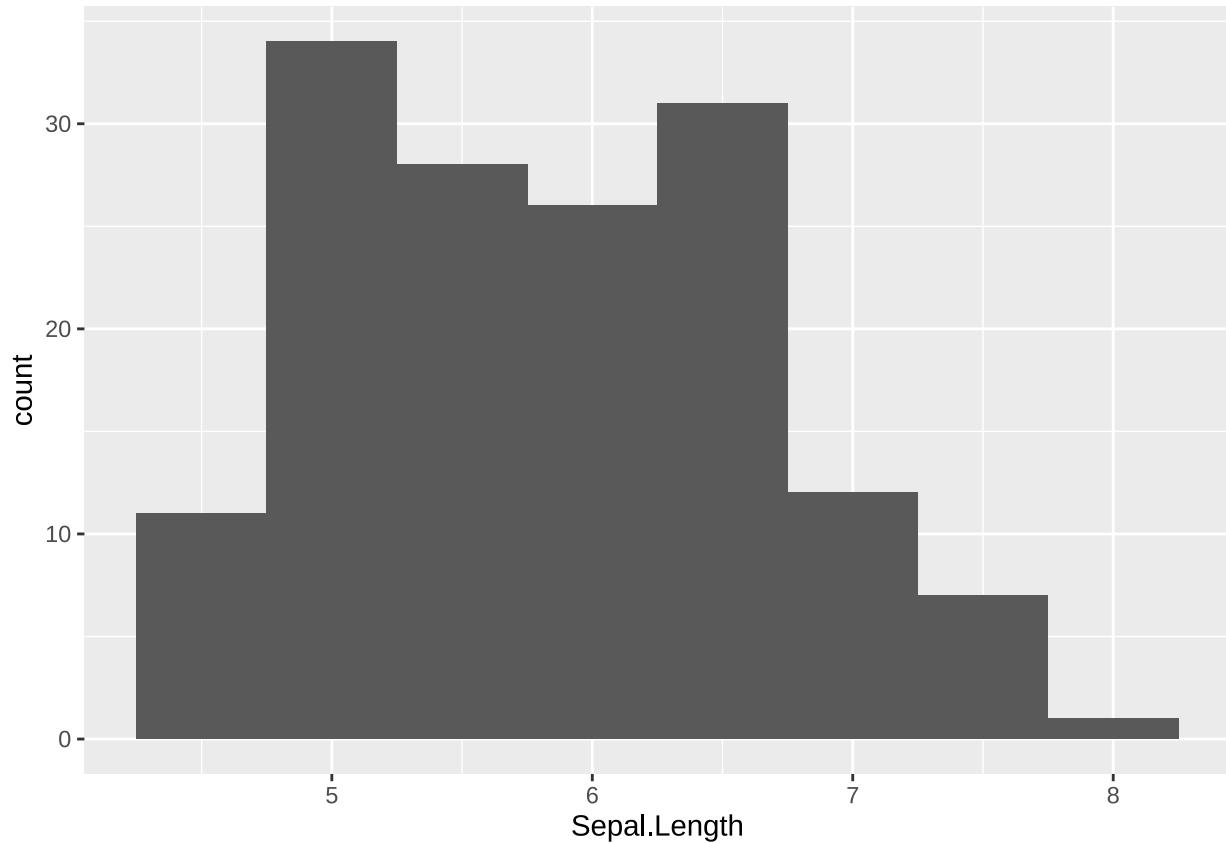
## 4.5 统计变换

概念：对数据所应用的统计类型/方法。

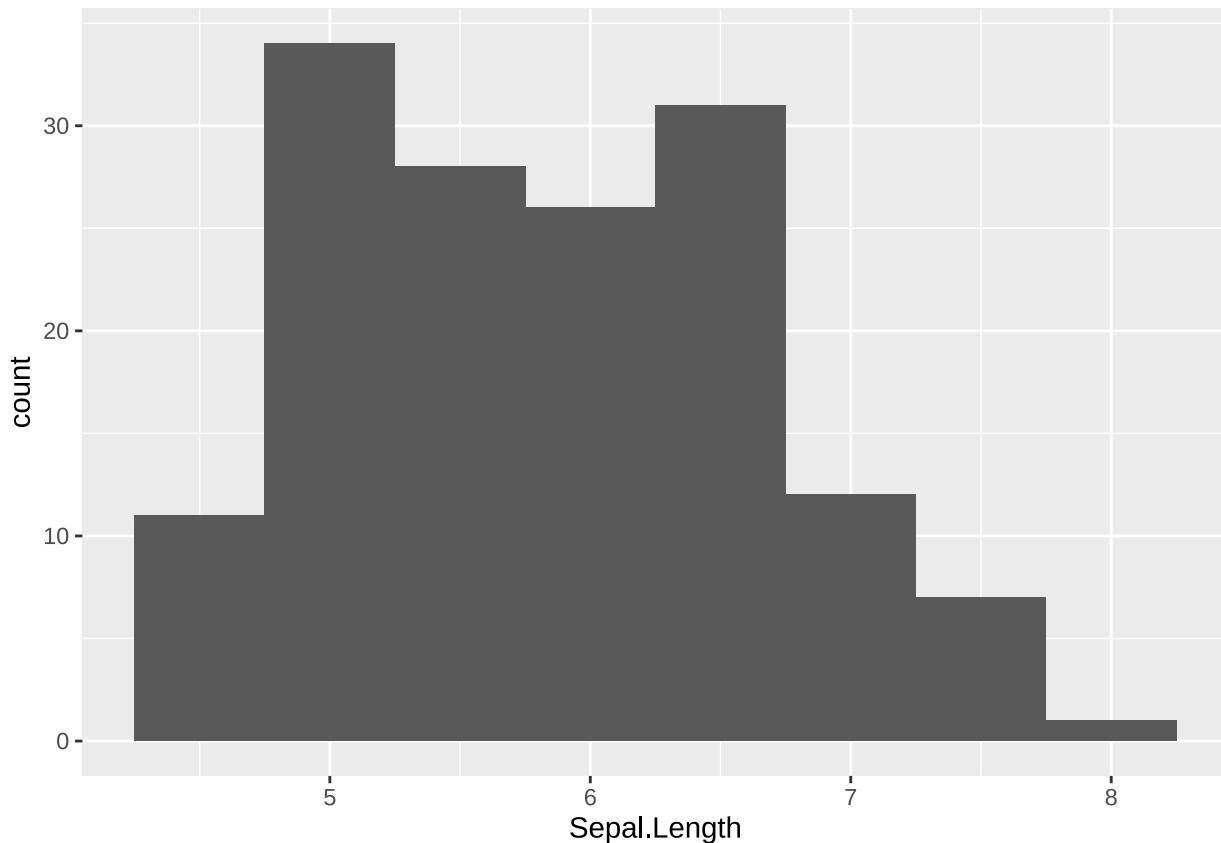
`ggplot2` 为每一种几何类型指定了一种默认的统计类型，如果仅指定 `geom` 或 `stat` 中的一个，另外一个会自动获取。其中，`stat_identity` 则表示不做任何的统计变换。

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)
```



## 4.6 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

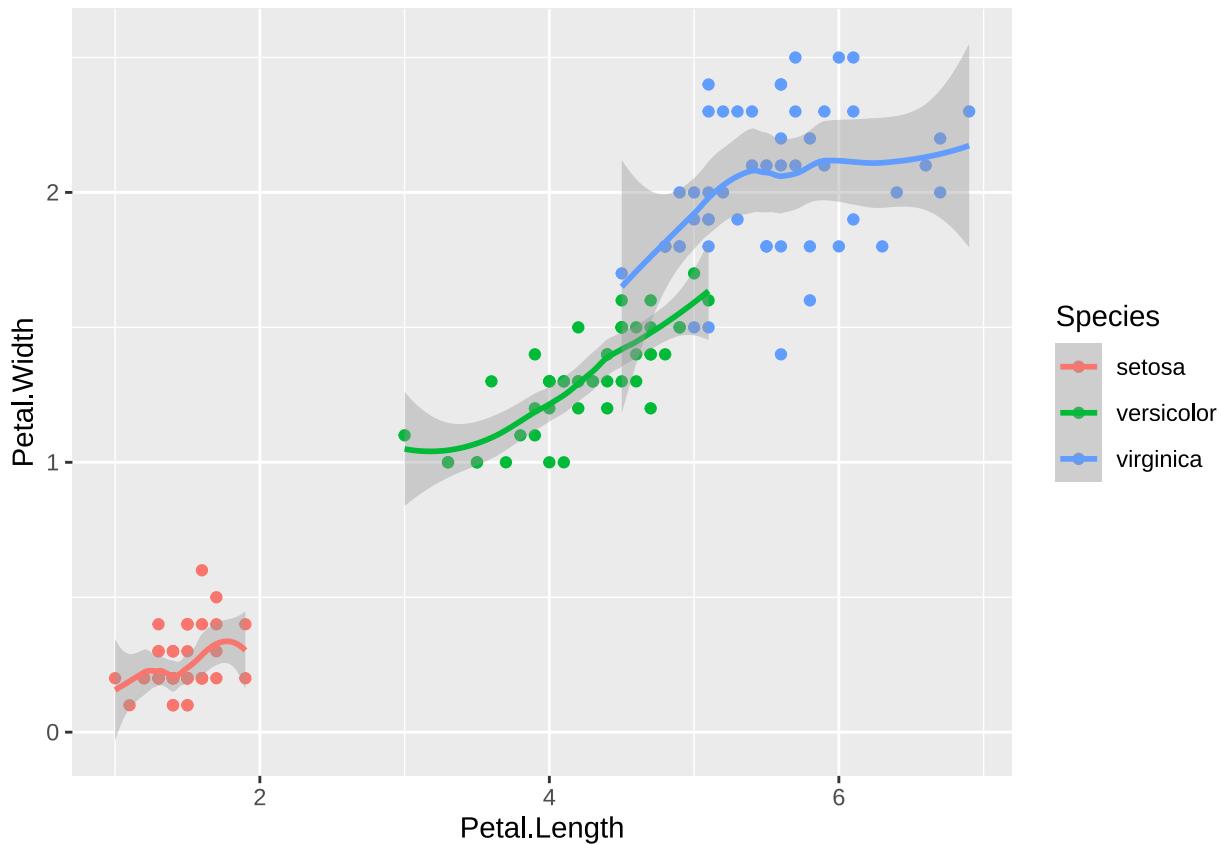
在 `ggplot()` 的 `mapping` 参数中指定 x 维、y 维、color 维等，实际上每一维度都有一个对应的默认刻度 (scale)，即，将数据值映射到图形中的映射方法。

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()
```



#### 4.6.1 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

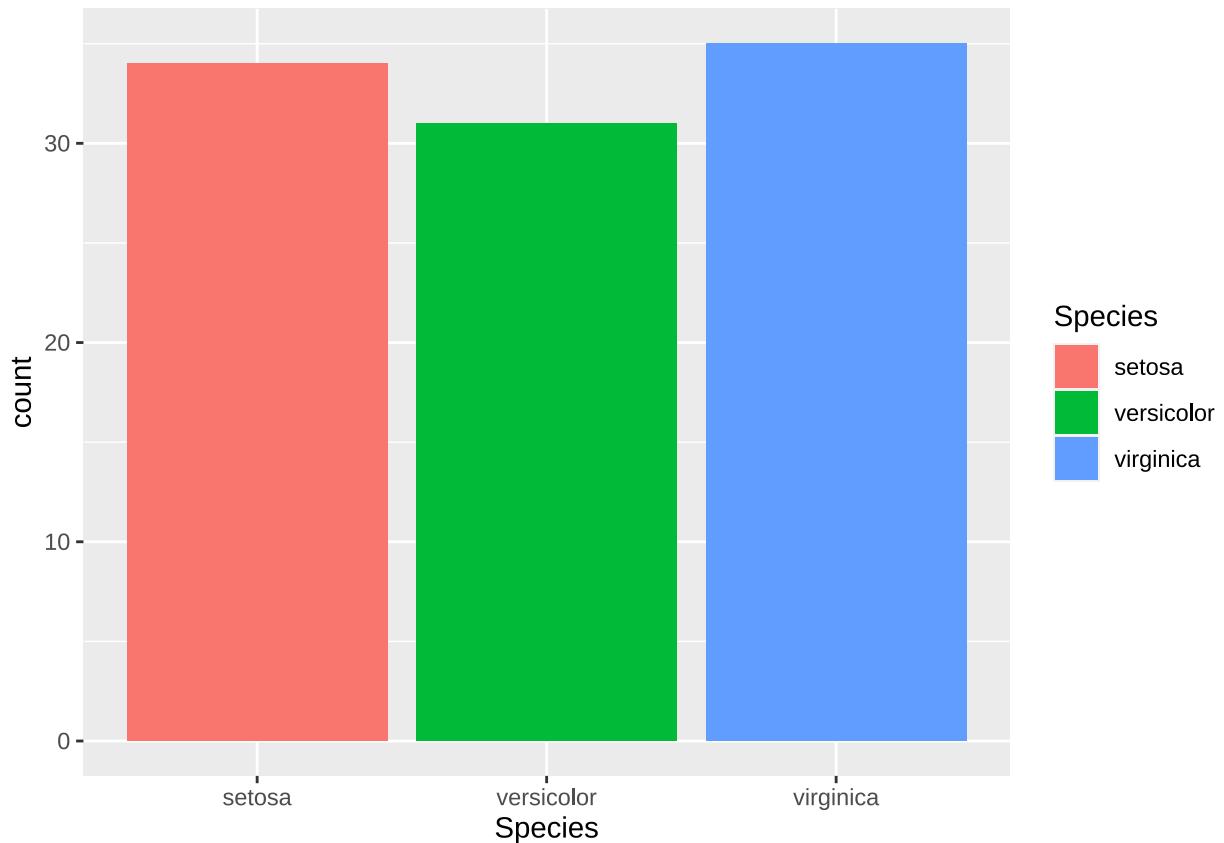
- `NULL`, 就是去掉标签
- 
- 参数 `values` 指的是（颜色、形状等）视觉属性值,
    - 要么, 与数值的顺序一致;
    - 要么, 与 `breaks` 提供的字符型向量长度一致
    - 要么, 用命名向量 `c(" 数据标签" = " 视觉属性")` 提供
- 
- 参数 `expand`, 控制参数溢出量
- 
- 参数 `range`, 设置尺寸大小范围, 比如针对点的相对大小

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。

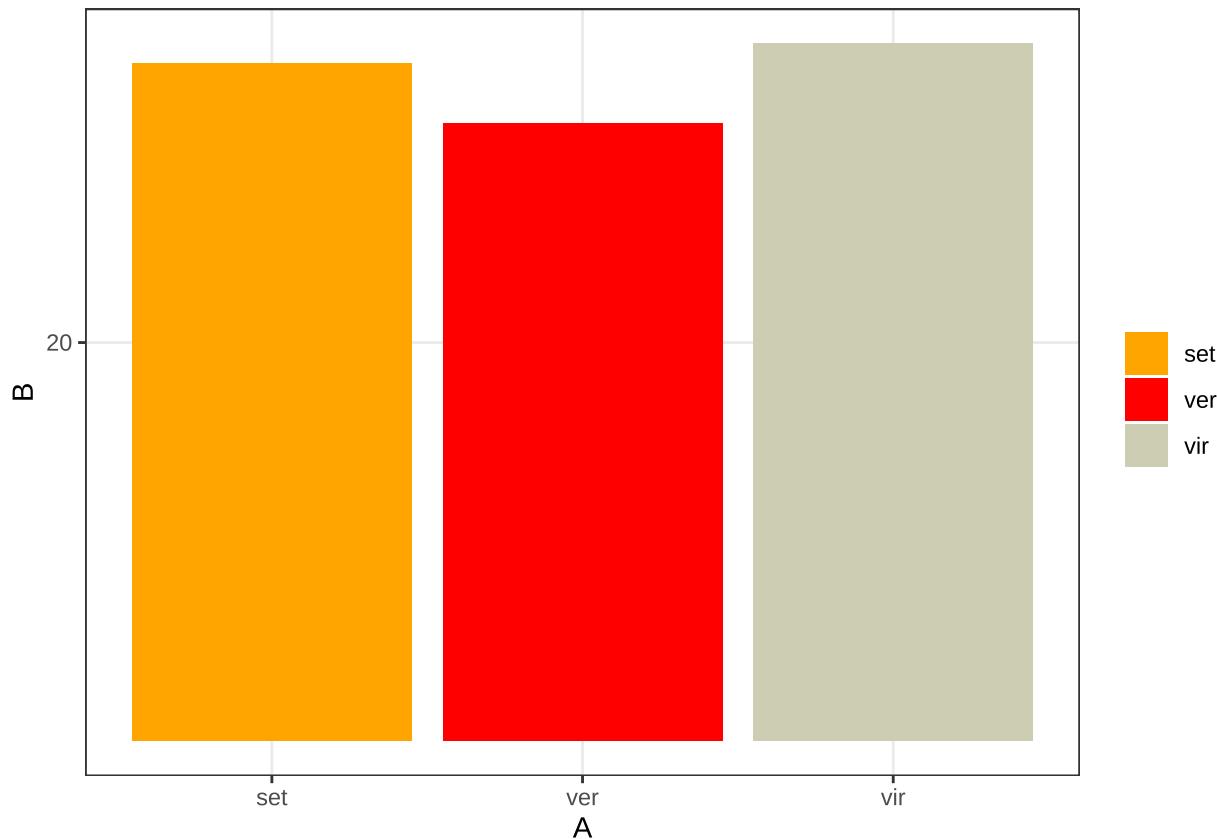
例子: 随机从 `iris` 数据集的 150 个样本中抽取 100 个样本, 并绘制条形图反映 100 个样本中各个鸢尾花种类的数量情况。然后通过修改标尺参数做前后对比图, 进而理解标尺在 `ggplot2` 包中的作用。

```
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
```

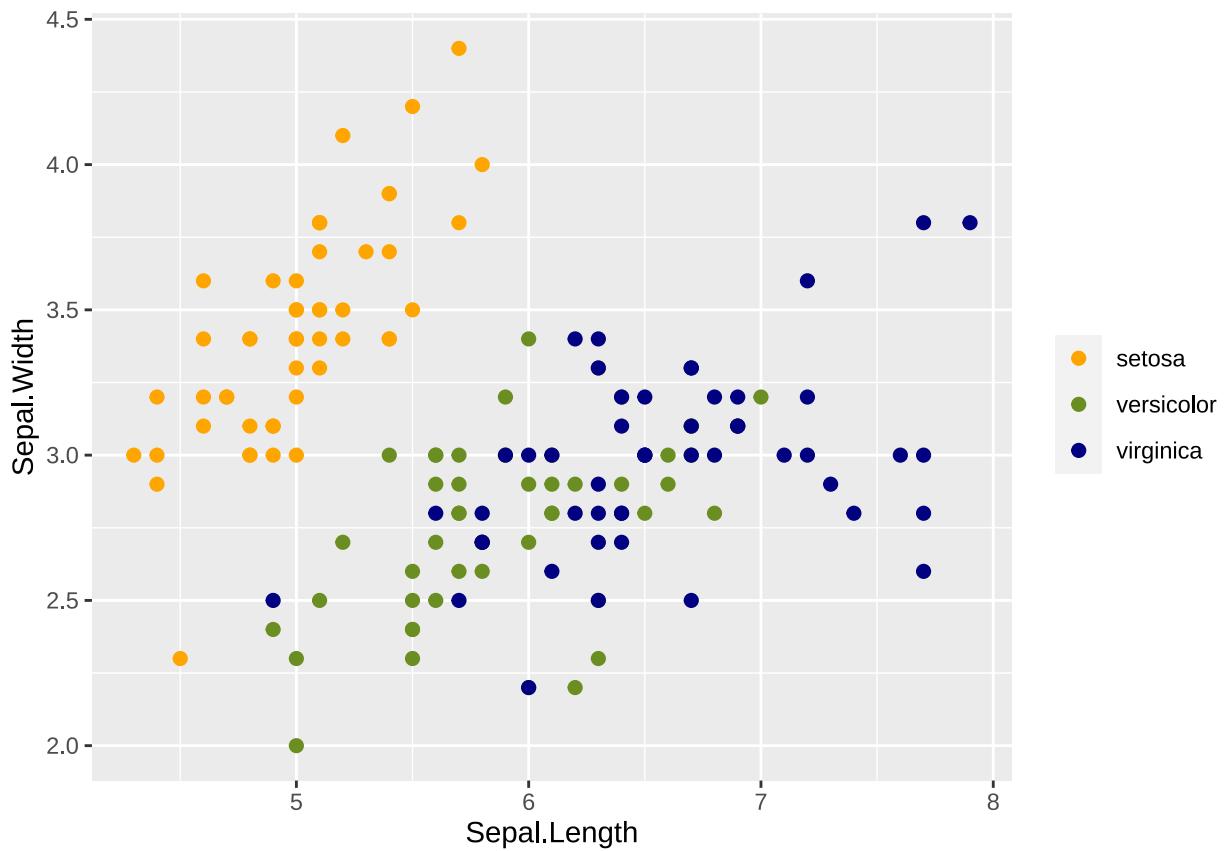


```
p + scale_fill_manual(  
  values = c("orange", "red", "lightyellow3"), # 颜色设置  
  name = NULL, # 图例和轴使用的名称  
  labels = c("set", "ver", "vir") # 图例使用的标签  
) +  
scale_x_discrete(labels = c("set", "ver", "vir"), name = "A") +  
scale_y_continuous(name = "B", breaks = c(20, 40)) +  
theme_bw()
```

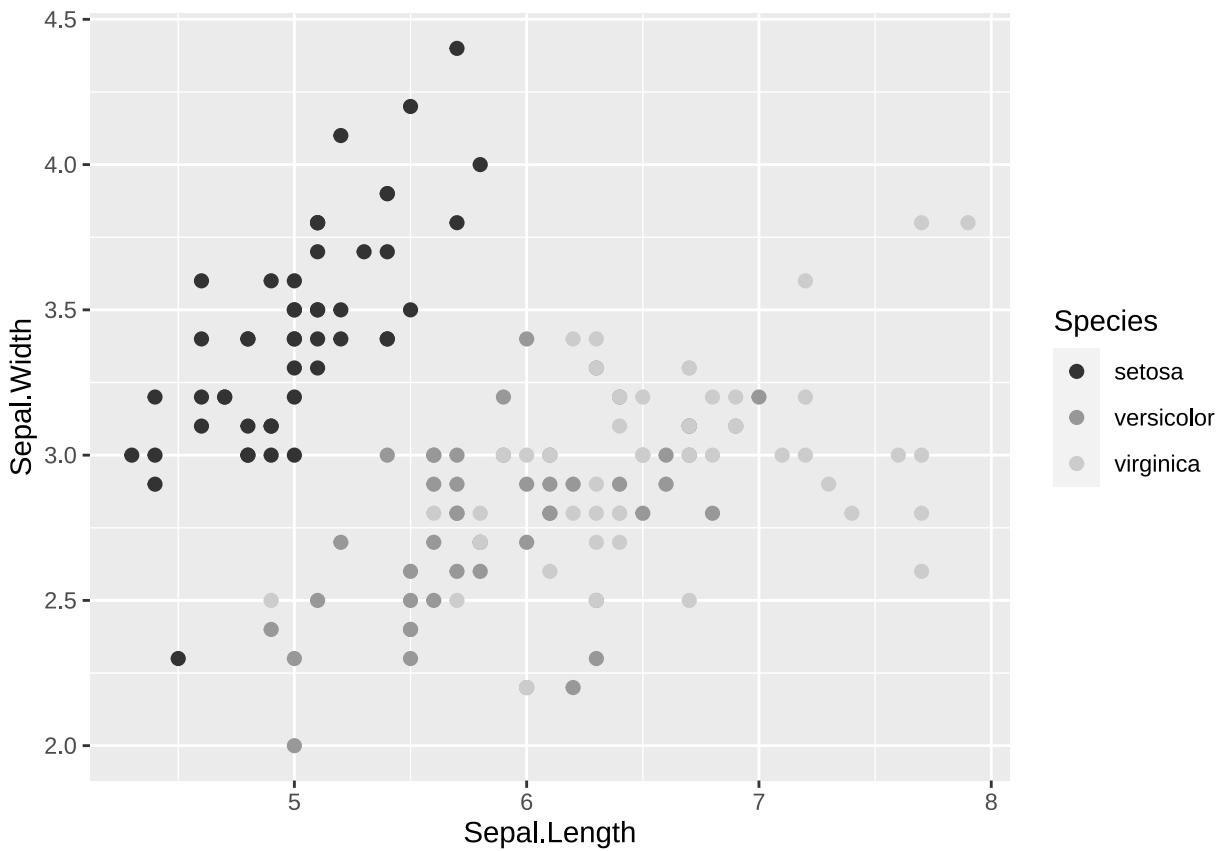


使用 `scale_color_manual` 或 `scale_color_brewer` 函数修改图形的颜色。在对 iris 数据集中的 Sepal.Length 与 Sepal.Width 的散点图分别使用以上两种方法修改散点颜色

```
# 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()
```

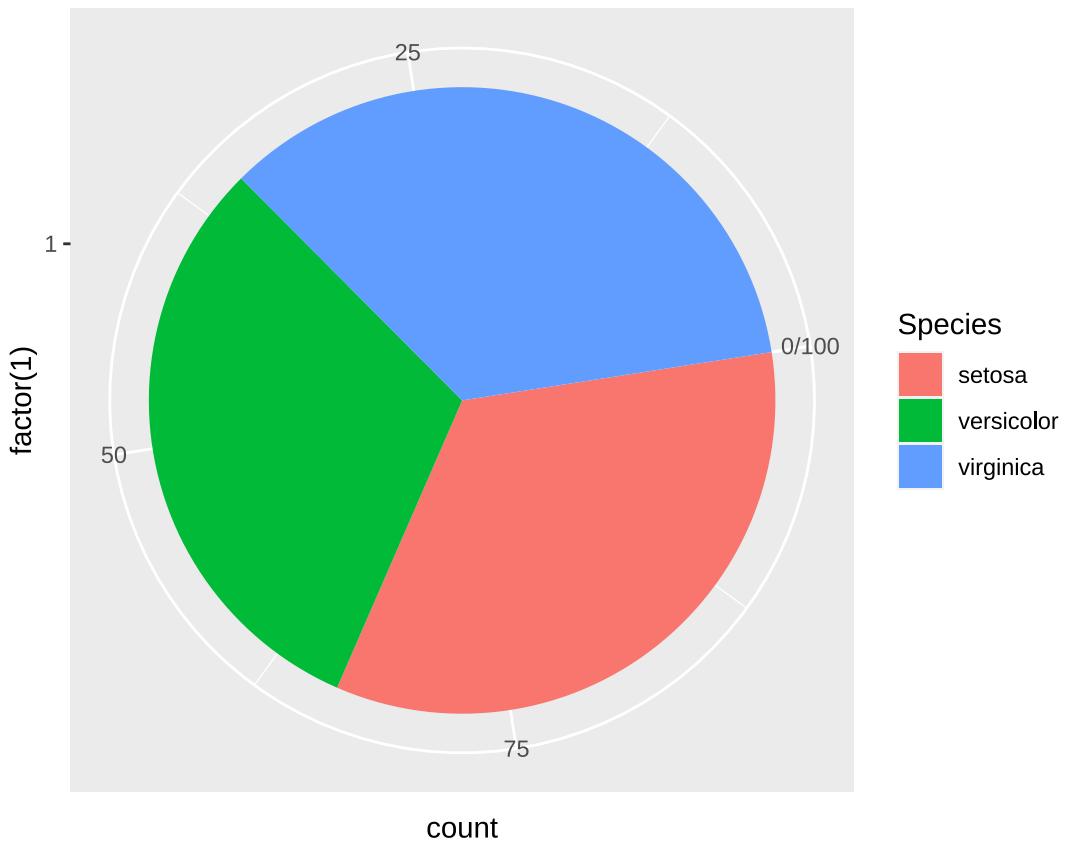
## 4.7 坐标系

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))`。

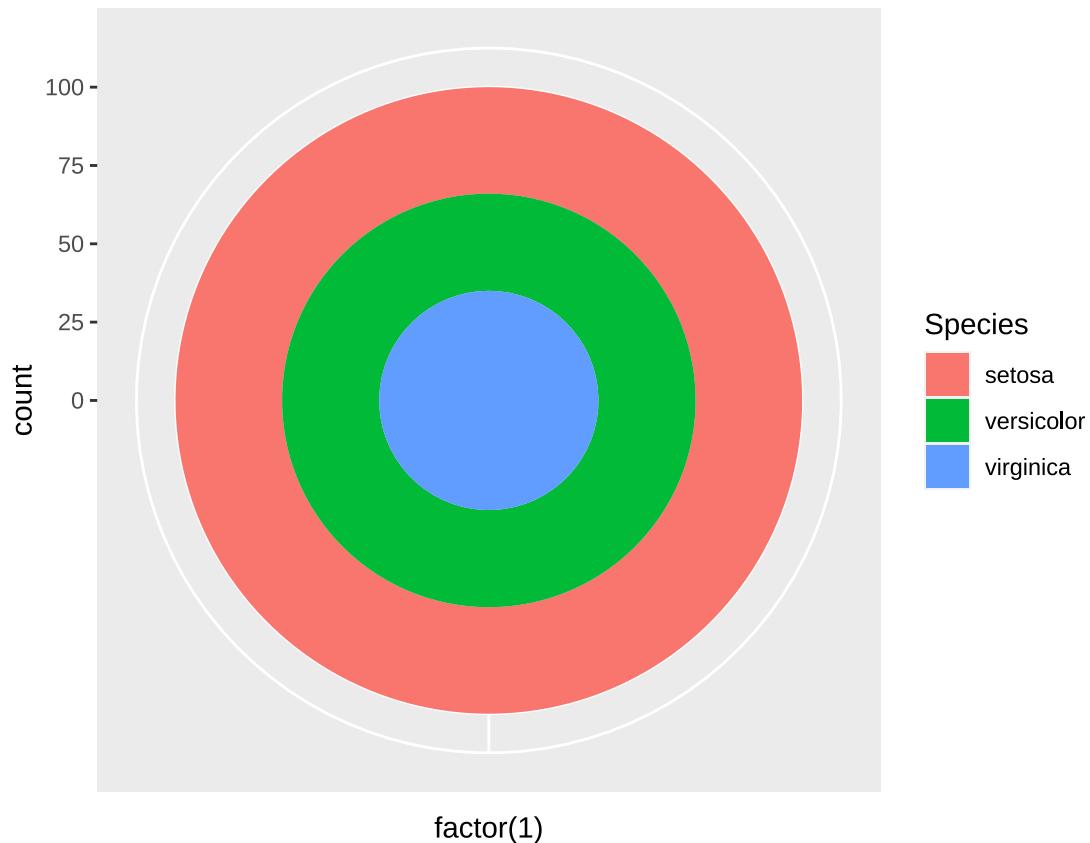
`coord_flip`: x 轴和 y 轴换位置。

`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。

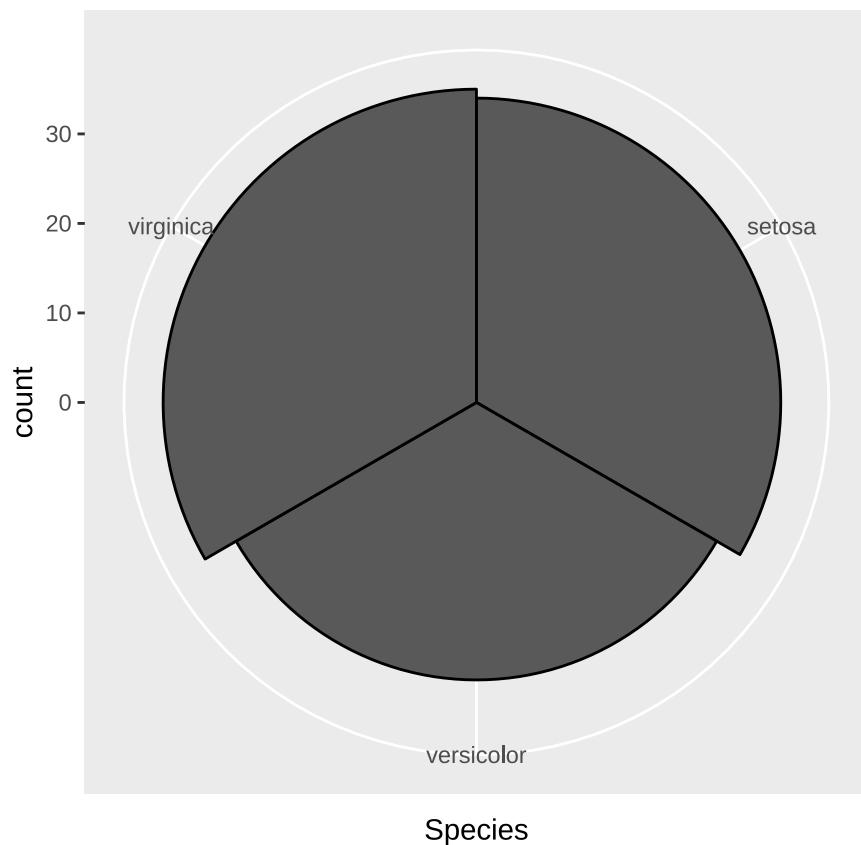
```
# 饼图 = 堆叠长条图 + polar_coordinates
pie <- ggplot(my_iris, aes(x = factor(1), fill = Species)) +
  geom_bar(width = 1)
pie + coord_polar(theta = "y", direction = -1, start = 30)
```



```
# 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()
```



## 4.8 分面

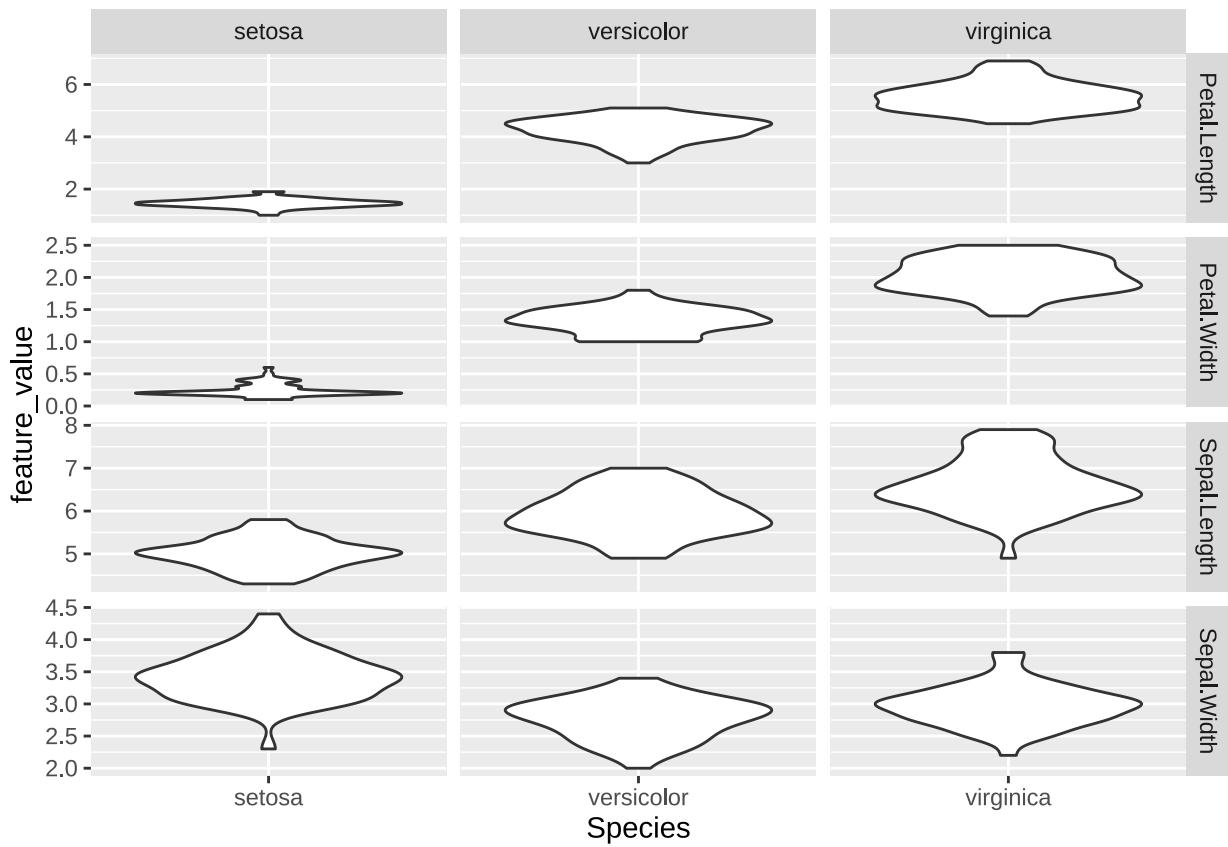
分面，就是分组绘图，根据定义的规则，将数据分为多个子集，每个子集按照统一的规则单独制图，排布在一个页面上。

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

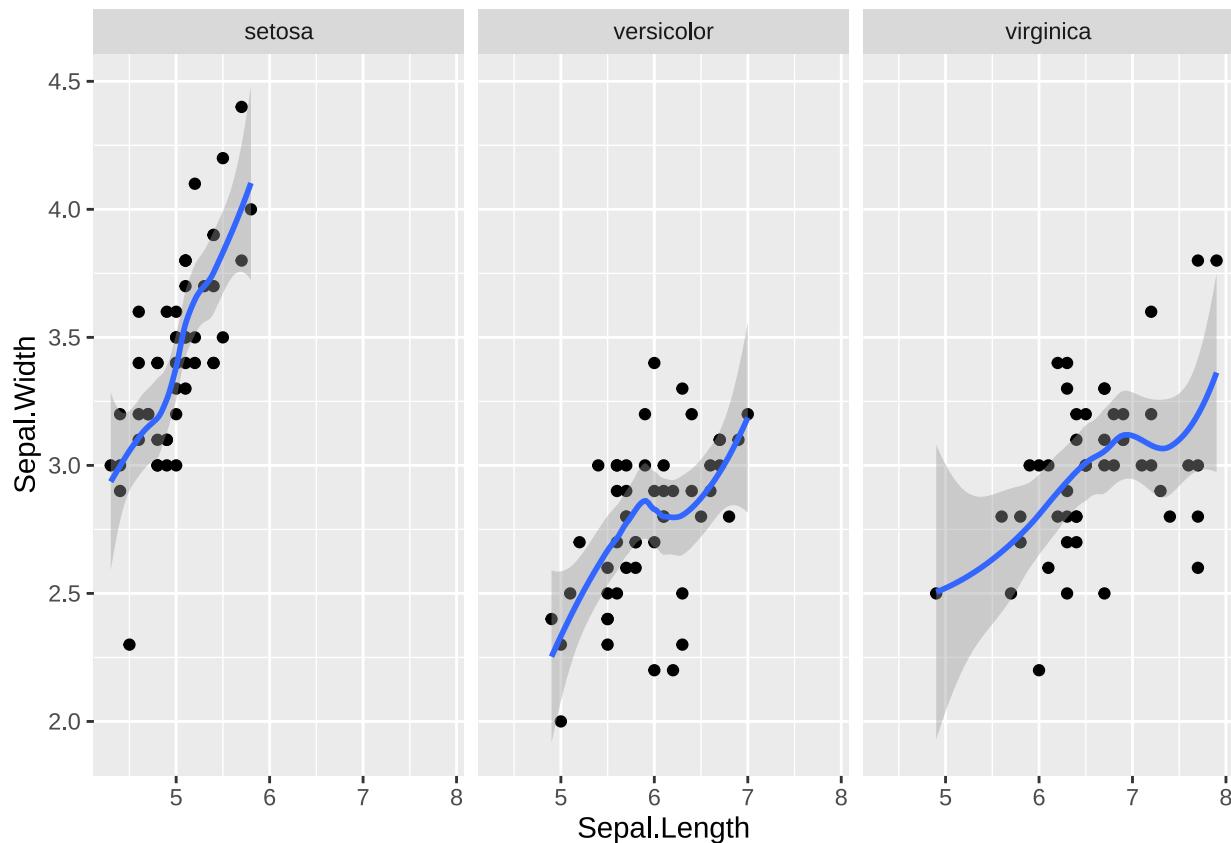
### 1. `facet_grid` 函数

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
```



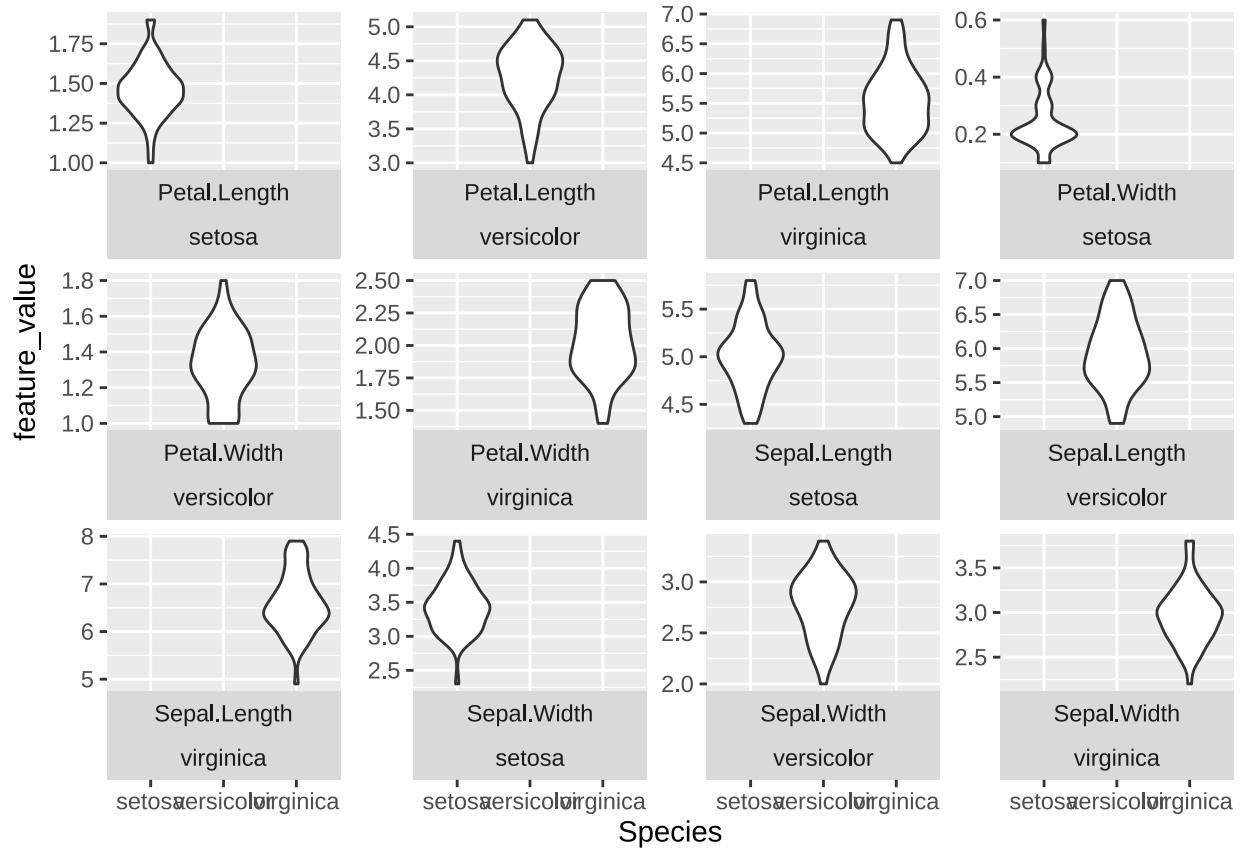
```
# iris 例子
ggplot(data = iris, mapping = aes(x = Sepal.Length, y = Sepal.Width)) + # 底层画布
  geom_point() +
  geom_smooth() +
  facet_grid(~Species)
```



## 2. facet\_wrap 函数

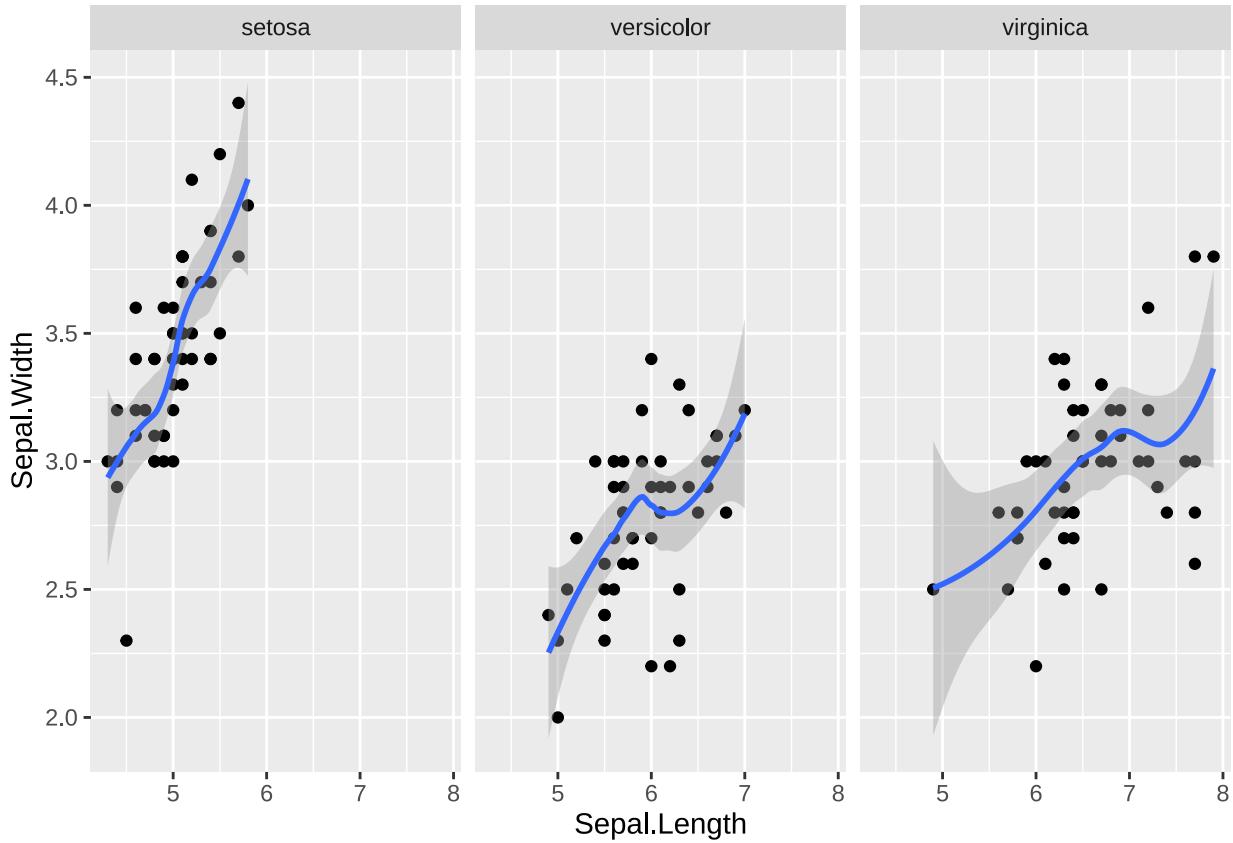
facet\_wrap 函数生成一个动态调整的一维布局，根据“~”位置变量 1+ 位置变量 2+...“来确定每个子集的位置，先逐行排列，放不下了移动到下一行。

```
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)
```



## 4.9 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。

### 4.9.1 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:

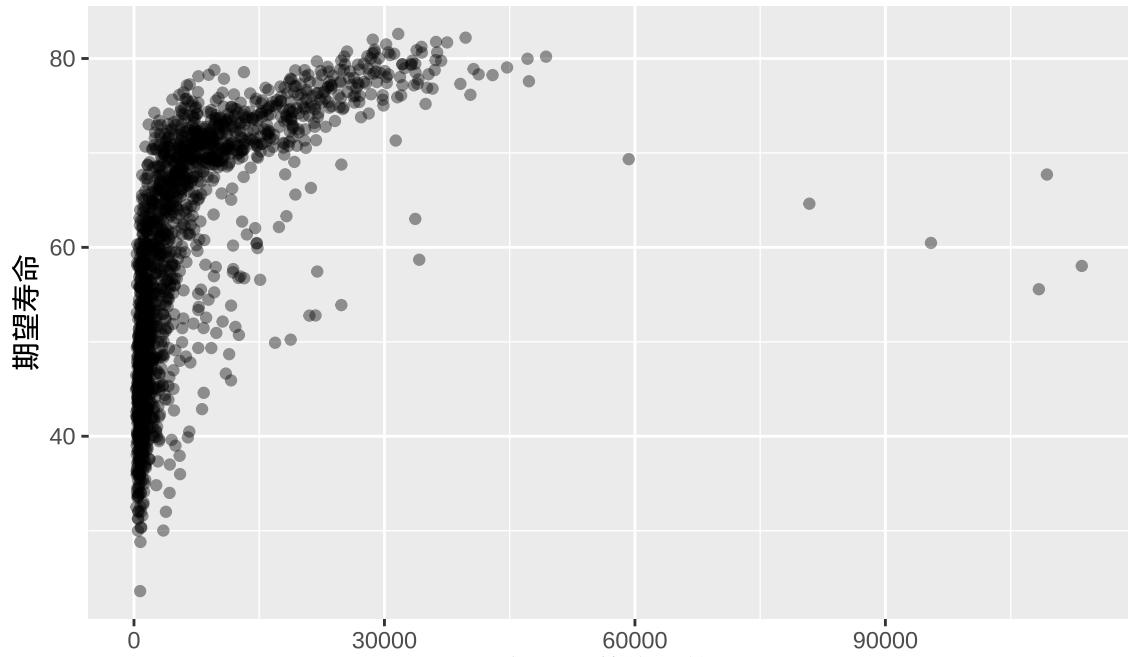
```
library(gapminder)
p <- ggplot(data = gapminder,
  mapping = aes(
    x = gdpPercap,
    y = lifeExp))
```

```
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

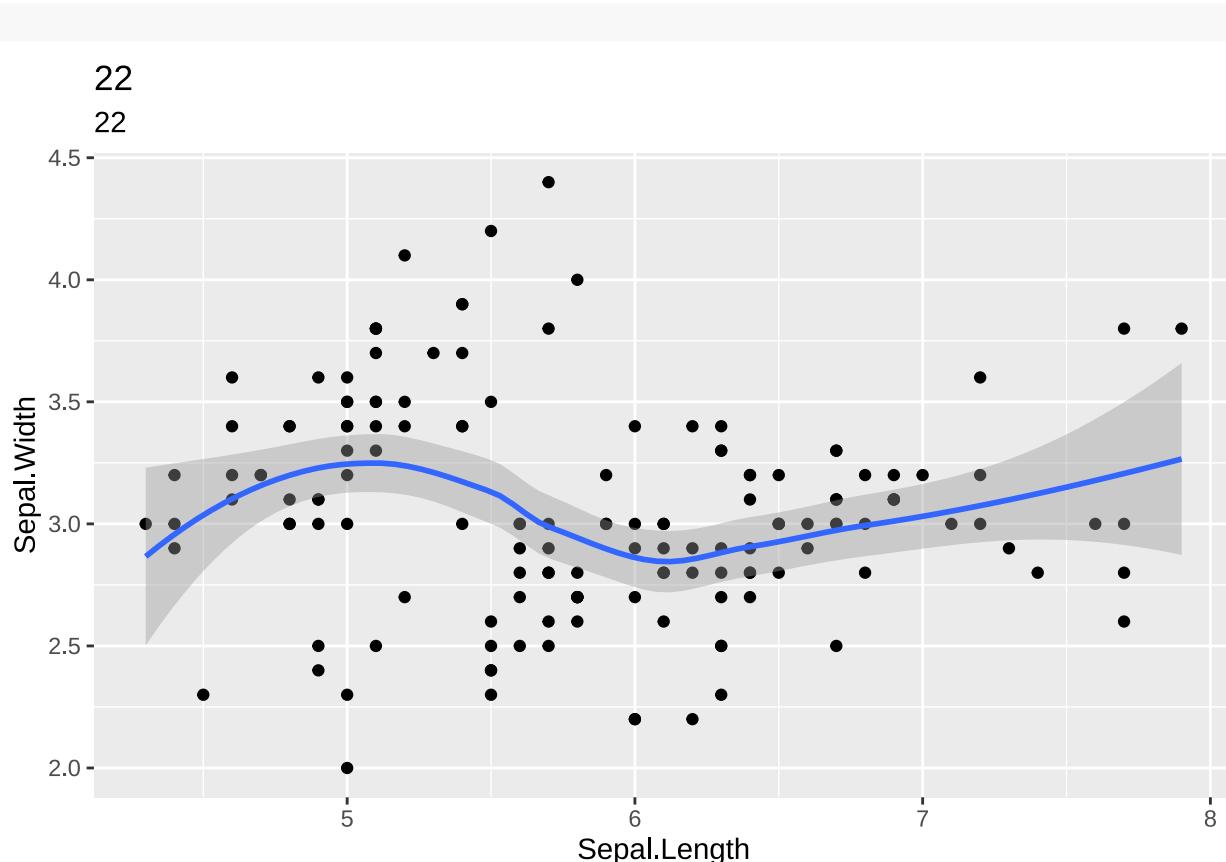
1952-2007



数据来源: gapminder

#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。

#### 4.9.2 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：

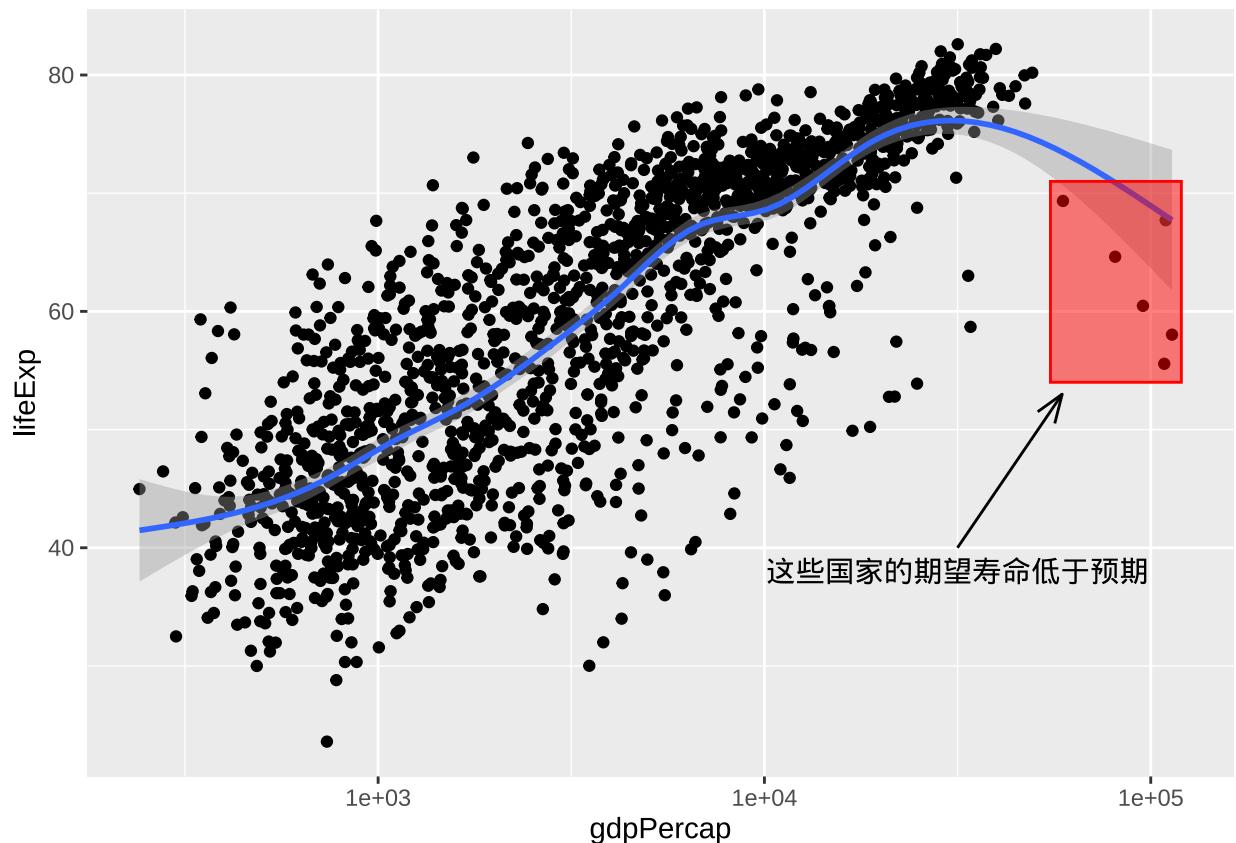
在 `annotate()` 中选 `geom="rect"`，给出长方形的左右和上限界限，可以将上面图形中最右侧偏低的点用长方形填充标出。可以在 `annotate()` 中选 `geom="line"` 画线，需要给出线的起点和终点坐标，可以 `arrow` 选项要求画箭头，用 `arrow()` 函数给出箭头的大小、角度等设置，如：

```
p + geom_point() +
  geom_smooth(method="gam") +
  scale_x_log10() +
  annotate(geom = "rect",
          xmin = 5.5E4, xmax = 1.2E5,
```

```

    ymin = 54, ymax = 71,col = 'red',fill = 'red',alpha = 0.5) +
annotate(geom = "line",
  x = c(5.9E4, 3.16E4),
  y = c(53, 40),
  arrow = arrow(angle = 20, length = unit(4, "mm")))) +
annotate(geom = "text",
  x = 3.16E4, y = 38,
  label = "这些国家的期望寿命低于预期")

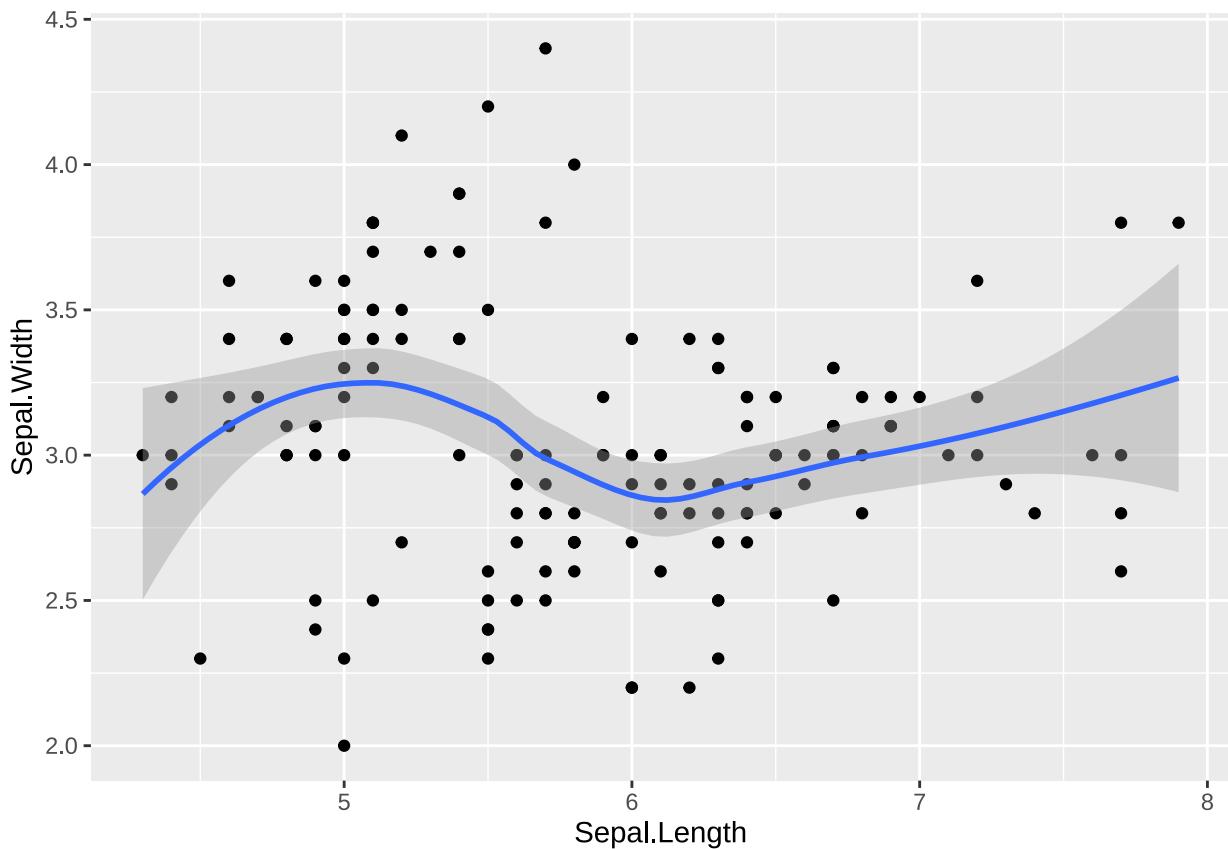
```



```

#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。

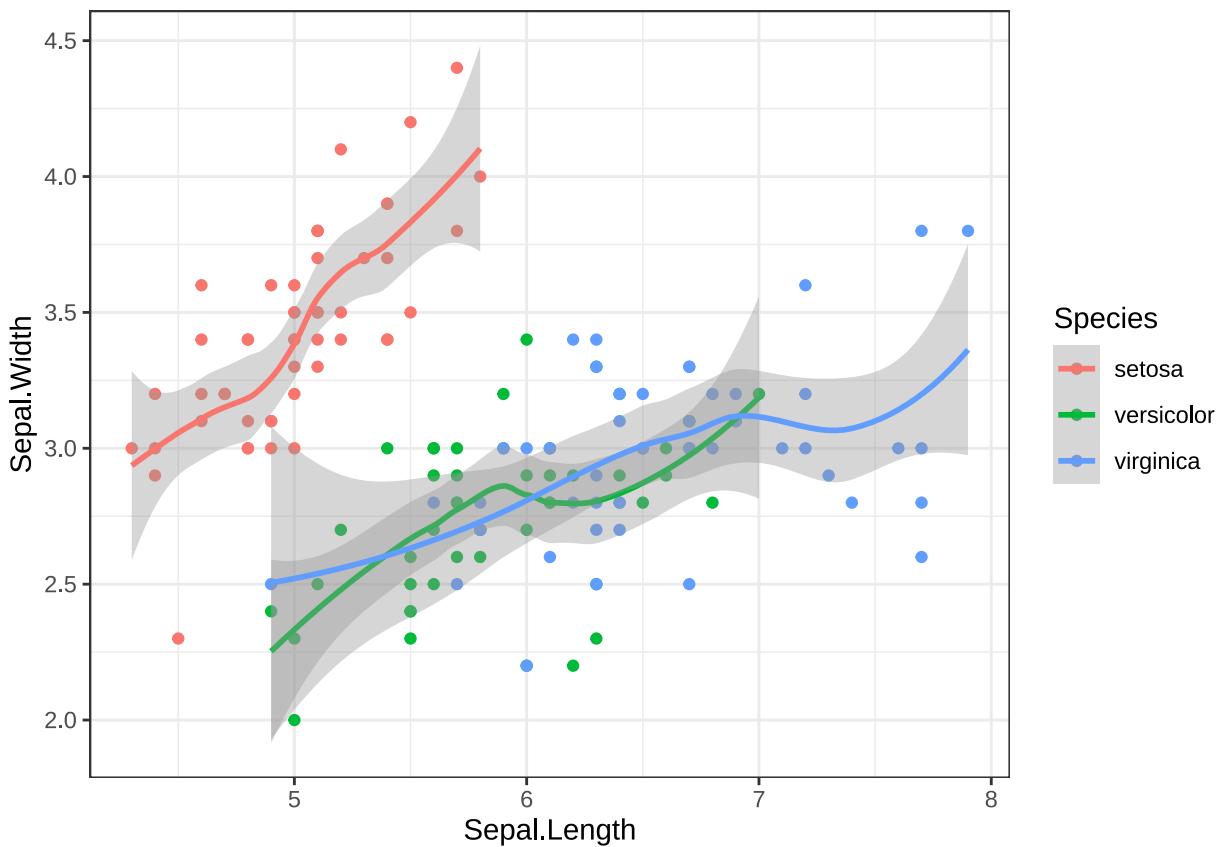
### 4.9.3 指南

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()
```



#### 4.9.4 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
```

## 4.10 保存图片

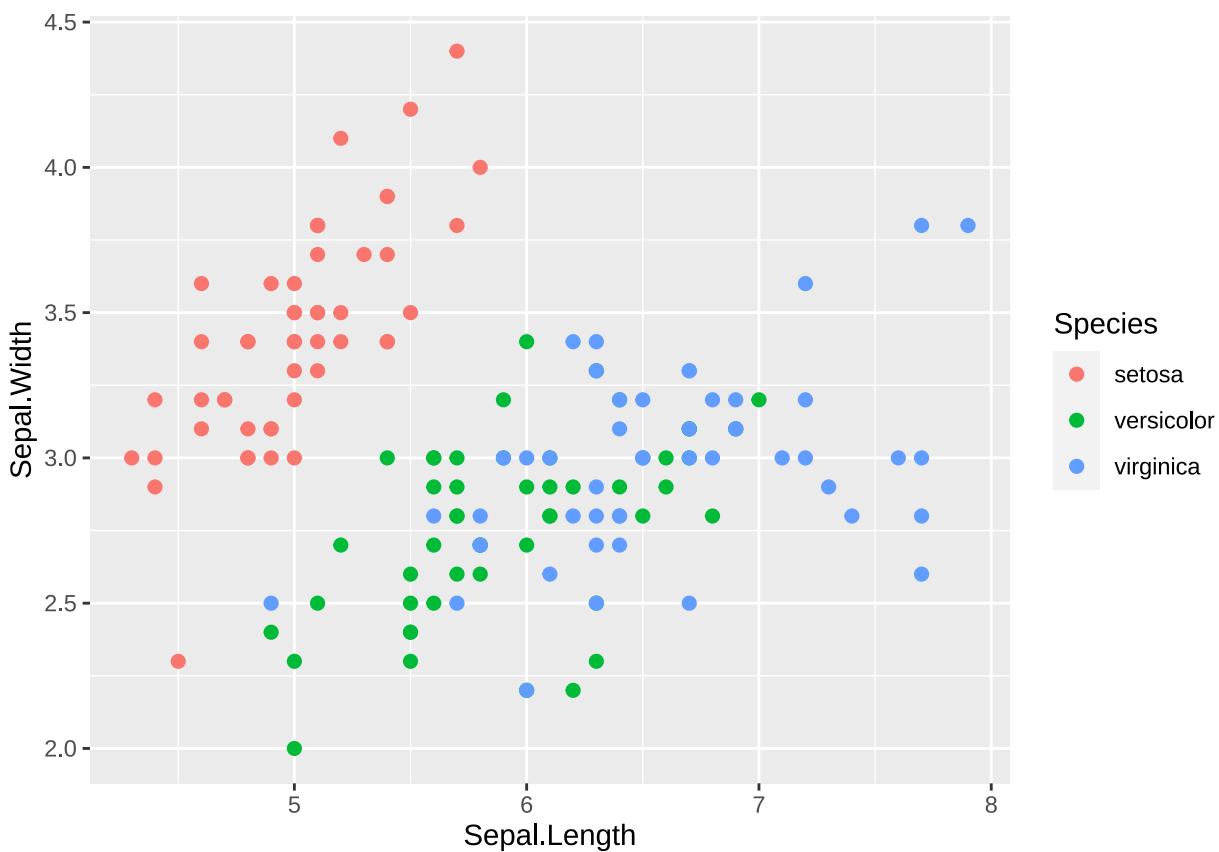
ggplot2 包中提供 ggsave 函数进行图形保存。ggsave 函数的使用格式如下所示。

```
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）

## 4.11 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。其他例子可以在第 ?? 找到。

### 4.11.1 柱状图 + 误差项

```
## 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

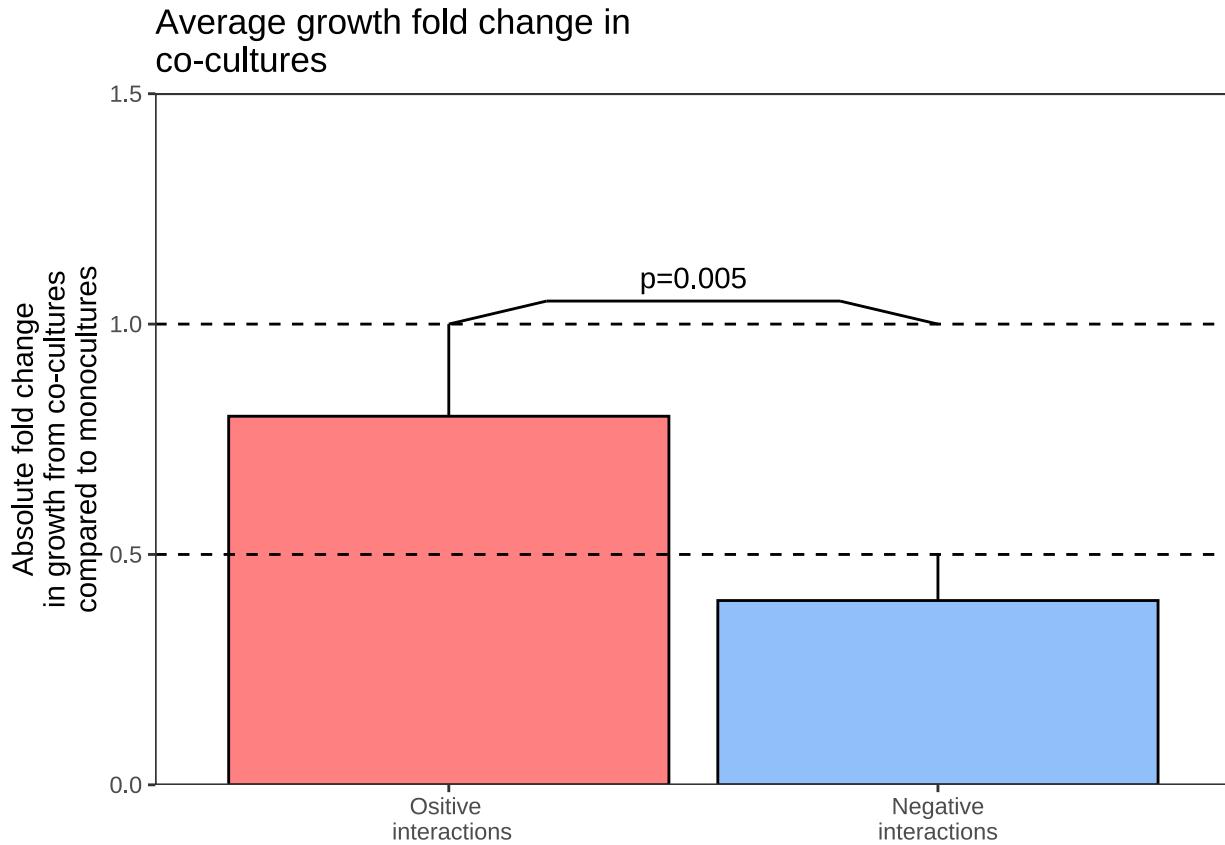
library(ggplot2)

#### 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\ninteractions")) + #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
```



#### 4.11.2 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)

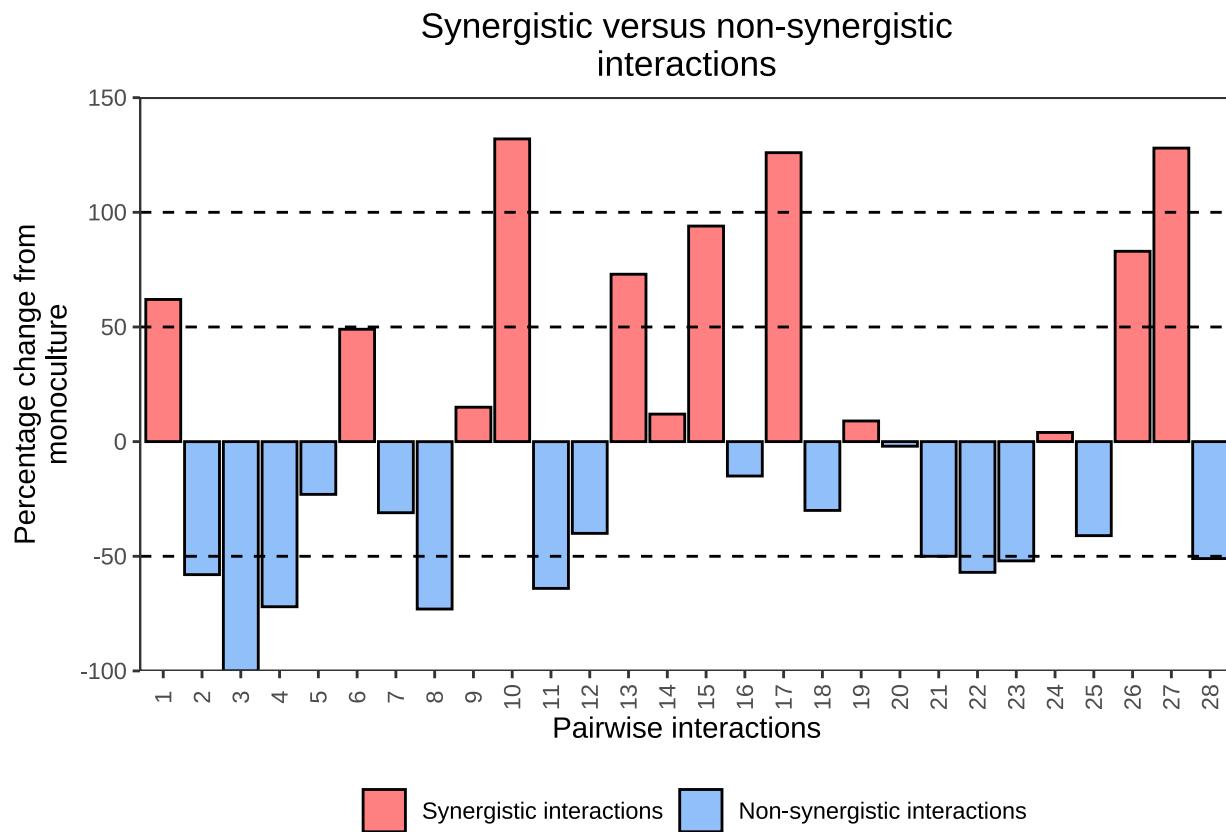
##      x      y          group
#> 1   1 -100 Non-synergistic interactions
#> 2   2  100 Non-synergistic interactions
#> 3   3   50 Synergistic interactions
#> 4   4   80 Synergistic interactions
#> 5   5   20 Non-synergistic interactions
```

```
##      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
```



### 4.11.3 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 5

## 常用图形

```
library(tidyverse)
library(ggExtra)
library(ragg)
library(ggalluvial)
library(treemapify)
library(ggalt)
library(palmerpenguins)

head(penguins)

## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex
##   <fct>    <fct>        <dbl>        <dbl>            <int>      <int> <fct>
## 1 Adelie   Torgo~       39.1         18.7            181      3750 male 
## 2 Adelie   Torgo~       39.5         17.4            186      3800 fema~
## 3 Adelie   Torgo~       40.3          18             195      3250 fema~
## 4 Adelie   Torgo~        NA           NA              NA       NA <NA>
## 5 Adelie   Torgo~       36.7         19.3            193      3450 fema~
## 6 Adelie   Torgo~       39.3         20.6            190      3650 male 
## # ... with 1 more variable: year <int>

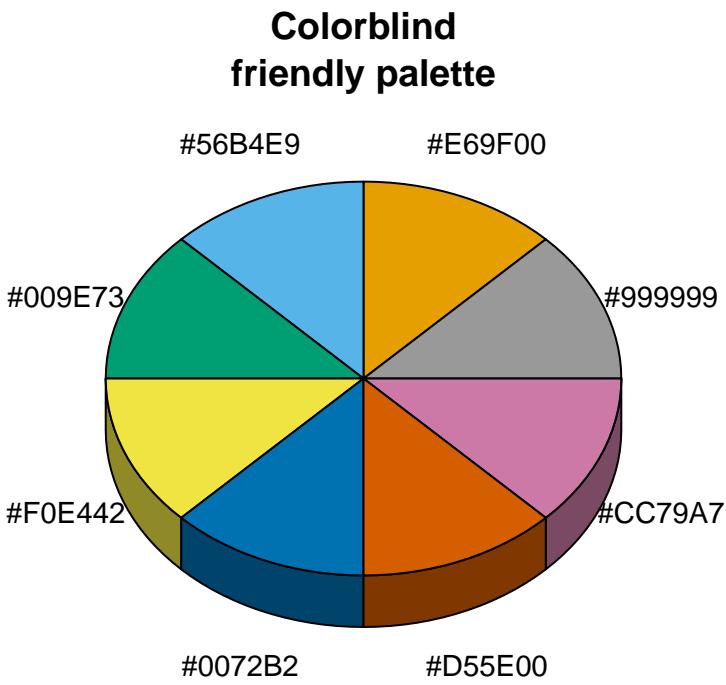
#head(penguins_raw)
```

- set colorblind-friendly palettes

```
# The palette with grey:
cbp1 <- c("#999999", "#E69F00", "#56B4E9", "#009E73",
          "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# The palette with black:
cbp2 <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
          "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

library(plotrix)
sliceValues <- rep(10, 8) # each slice value=10 for proportionate slices
(
  p <- pie3D(sliceValues,
             explode=0,
             theta = 1.2,
             col = cbp1,
             labels = cbp1,
             labelcex = 0.9,
             shade = 0.6,
             main = "Colorblind\nfriendly palette")
)
```



```
## [1] 0.3926991 1.1780972 1.9634954 2.7488936 3.5342917 4.3196899 5.1050881
```

```

## [8] 5.8904862

ggplot <- function(...) ggplot2::ggplot(...) +
  scale_color_manual(values = cbp1) +
  scale_fill_manual(values = cbp1) # note: needs to be overridden when using continuous color scale
  theme_bw()

penguins

## # A tibble: 344 x 8
##   species island   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##   <fct>   <fct>        <dbl>        <dbl>          <int>        <int>
## 1 Adelie  Torgersen     39.1       18.7           181        3750
## 2 Adelie  Torgersen     39.5       17.4           186        3800
## 3 Adelie  Torgersen     40.3        18            195        3250
## 4 Adelie  Torgersen      NA         NA             NA         NA
## 5 Adelie  Torgersen     36.7       19.3           193        3450
## 6 Adelie  Torgersen     39.3       20.6           190        3650
## 7 Adelie  Torgersen     38.9       17.8           181        3625
## 8 Adelie  Torgersen     39.2       19.6           195        4675
## 9 Adelie  Torgersen     34.1       18.1           193        3475
## 10 Adelie Torgersen      42         20.2           190        4250
## # ... with 334 more rows, and 2 more variables: sex <fct>, year <int>

penguins_long <- penguins %>%
  gather("key", "value", bill_length_mm:body_mass_g)

penguins %>%
  group_by(species) %>%
  count(sex) %>%
  arrange(desc(n))

## # A tibble: 8 x 3
## # Groups:   species [3]
##   species   sex     n
##   <fct>   <fct> <int>
## 1 Adelie female    73
## 2 Adelie male     73
## 3 Gentoo male     61
## 4 Gentoo female   58
## 5 Chinstrap female 34

```

```

## 6 Chinstrap male      34
## 7 Adelie   <NA>      6
## 8 Gentoo  <NA>      5

penguins %>%
  gather("key", "value", bill_length_mm:body_mass_g) %>%
  group_by(species, sex, island, key) %>%
  summarise(n = n(),
            sum = sum(value),
            mean = mean(value),
            median = median(value),
            sd = sd(value),
            se = sd(value) / sqrt(n()))
}

## # A tibble: 52 x 10
## # Groups:   species, sex, island [13]
##   species sex   island   key       n     sum    mean median     sd     se
##   <fct>   <fct> <fct>   <chr>   <int>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Adelie  female Biscoe bill_dept~    22    389.   17.7   17.7   1.09   0.233
## 2 Adelie  female Biscoe bill_leng~    22    822.   37.4   37.8   1.76   0.376
## 3 Adelie  female Biscoe body_mass~    22 74125. 3369.  3375   343.   73.2
## 4 Adelie  female Biscoe flipper_l~    22   4118.  187.   187    6.74   1.44
## 5 Adelie  female Dream  bill_dept~    27    476.   17.6   17.8   0.897  0.173
## 6 Adelie  female Dream  bill_leng~    27    997.   36.9   36.8   2.09   0.402
## 7 Adelie  female Dream  body_mass~    27 90300. 3344.  3400   212.   40.8
## 8 Adelie  female Dream  flipper_l~    27   5072.  188.   188    5.51   1.06
## 9 Adelie  female Torgersen bill_dept~  24    421.   17.6   17.4   0.880  0.180
## 10 Adelie  female Torgersen bill_leng~  24    901.   37.6   37.6   2.21   0.451
## # ... with 42 more rows

penguins %>%
  group_by(species, sex, island, year) %>%
  summarise_each(funs(sum,
                      mean,
                      median,
                      sd,
                      se = sd(.) / sqrt(n())))
)

```

```

## # A tibble: 35 x 24
## # Groups:   species, sex, island [13]
##   species sex   island year bill_length_mm_~ bill_depth_mm_s~ flipper_length_~
##   <fct>   <fct> <fct>    <dbl>           <dbl>           <int>
## 1 Adelie  fema~ Biscoe  2007     187.          92.9        909
## 2 Adelie  fema~ Biscoe  2008     330.          155        1679
## 3 Adelie  fema~ Biscoe  2009     305.          142.        1530
## 4 Adelie  fema~ Dream   2007     341.          161.        1665
## 5 Adelie  fema~ Dream   2008     290.          142.        1512
## 6 Adelie  fema~ Dream   2009     366.          173.        1895
## 7 Adelie  fema~ Torge~  2007     306.          145.        1501
## 8 Adelie  fema~ Torge~  2008     293.          139.        1520
## 9 Adelie  fema~ Torge~  2009     302.          137.        1498
## 10 Adelie male   Biscoe  2007    196.          91.5        908
## # ... with 25 more rows, and 17 more variables: body_mass_g_sum <int>,
## #   bill_length_mm_mean <dbl>, bill_depth_mm_mean <dbl>,
## #   flipper_length_mm_mean <dbl>, body_mass_g_mean <dbl>,
## #   bill_length_mm_median <dbl>, bill_depth_mm_median <dbl>,
## #   flipper_length_mm_median <dbl>, body_mass_g_median <dbl>,
## #   bill_length_mm_sd <dbl>, bill_depth_mm_sd <dbl>,
## #   flipper_length_mm_sd <dbl>, body_mass_g_sd <dbl>, ...

```

## 5.1 散点图

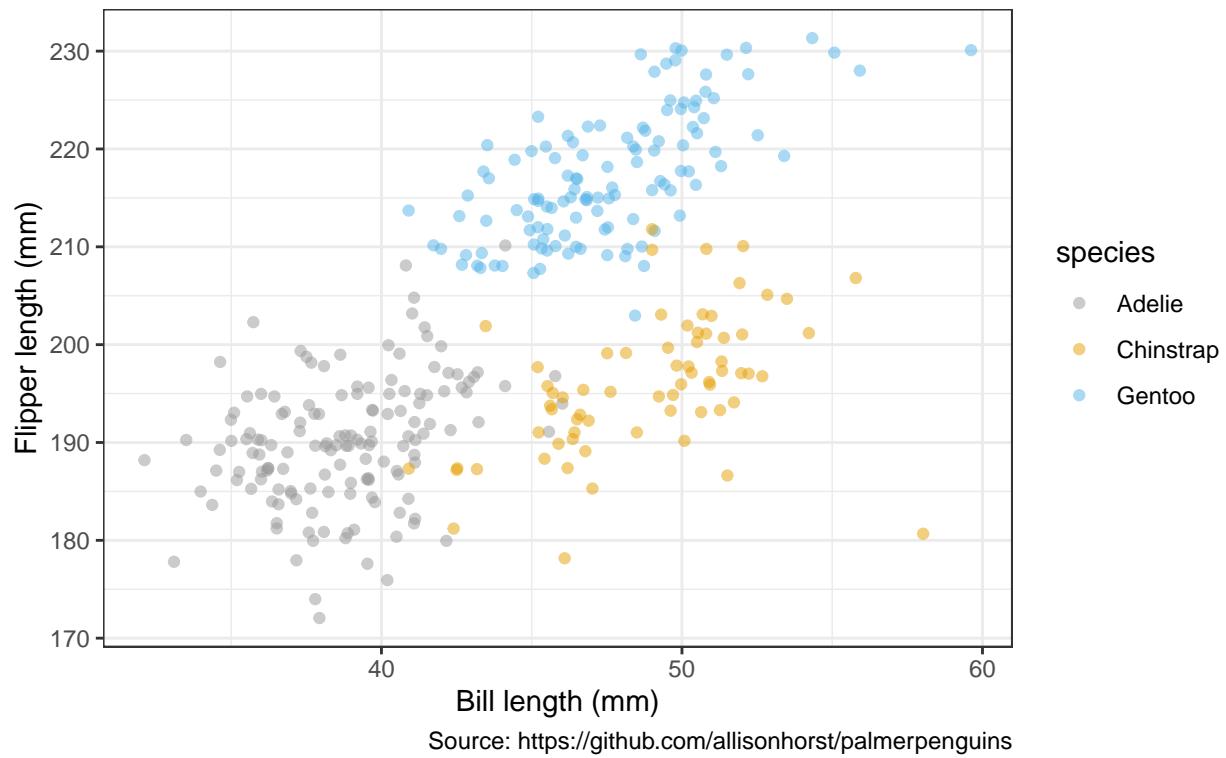
```

penguins %>%
  remove_missing() %>%
  ggplot(aes(x = bill_length_mm, y = flipper_length_mm, color = species)) +
  geom_jitter(alpha = 0.5) +
  #facet_wrap(vars(species), ncol = 3) +
  #scale_x_reverse() +
  #scale_y_reverse() +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       size = "body mass (g)",
       title = "Scatterplot",
       subtitle = "Penguins bill v. flipper length by species",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")

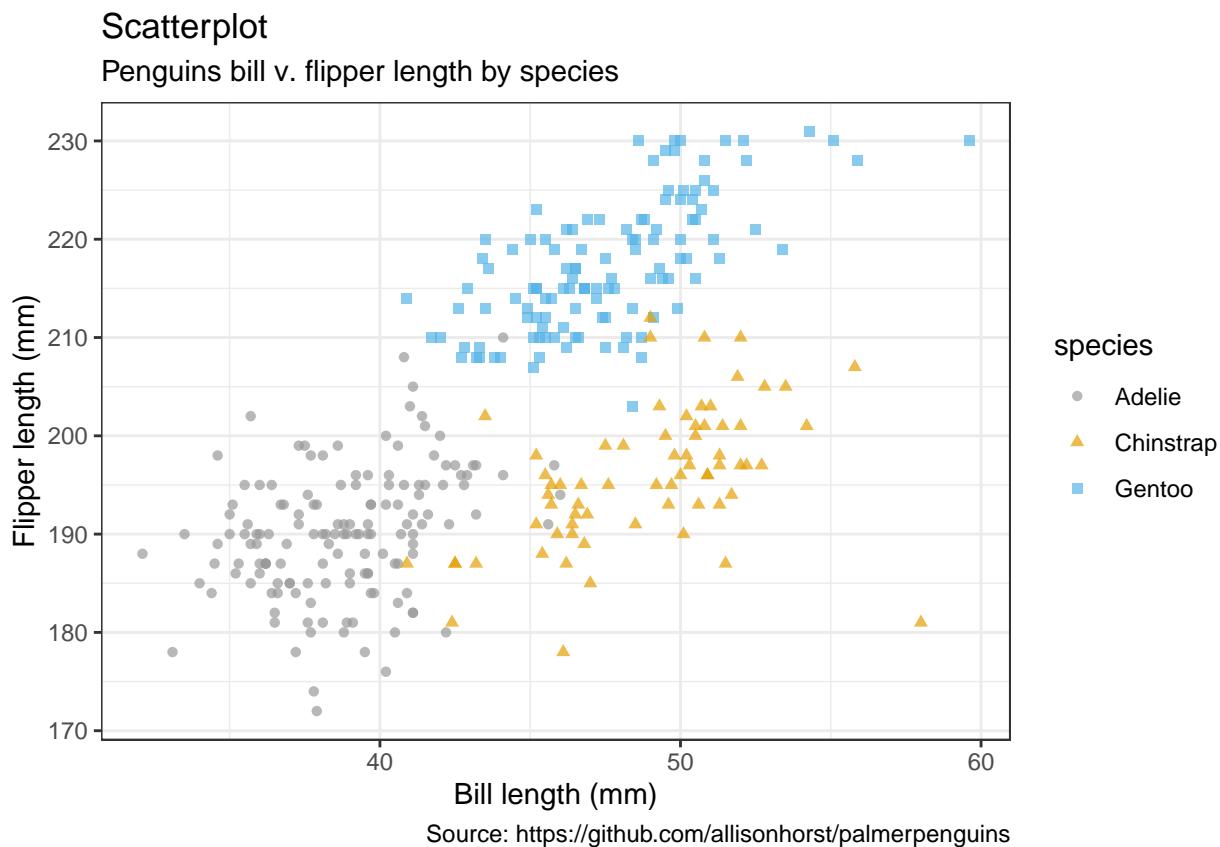
```

## Scatterplot

Penguins bill v. flipper length by species



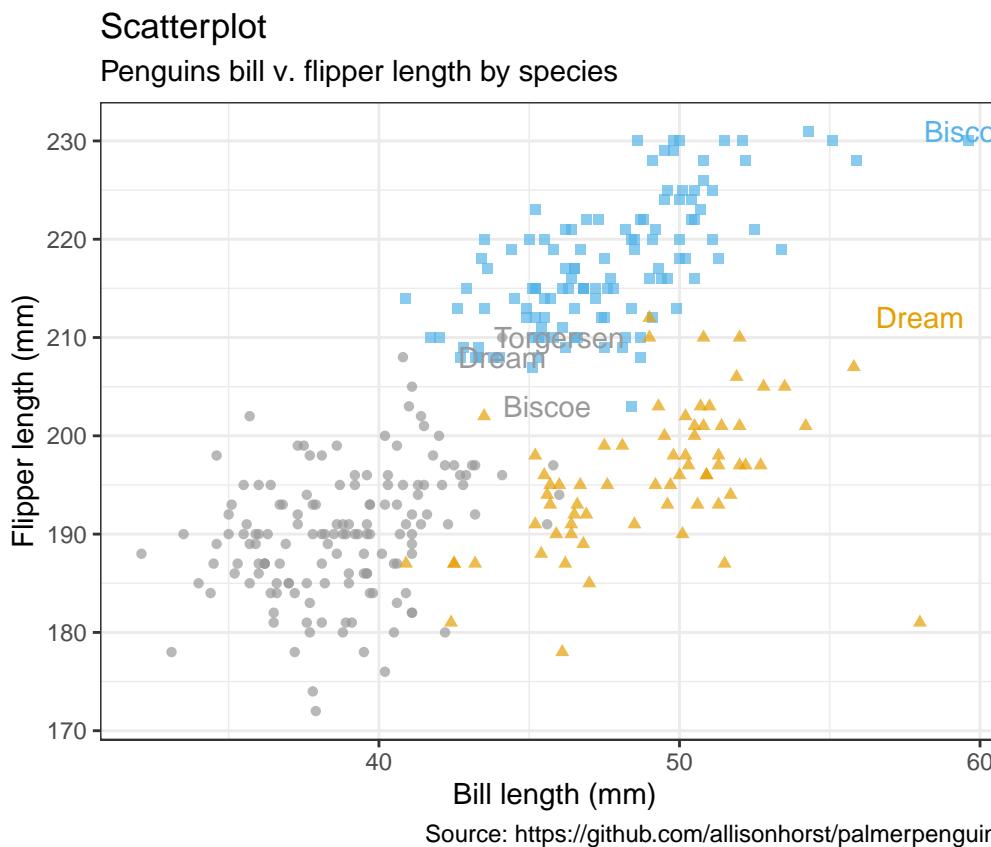
```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = bill_length_mm, y = flipper_length_mm,
             color = species, shape = species)) +
  geom_point(alpha = 0.7) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Scatterplot",
       subtitle = "Penguins bill v. flipper length by species",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```



- with labels/text

```
max_labels <- penguins %>%
  remove_missing() %>%
  group_by(species, island) %>%
  summarise(bill_length_mm = max(bill_length_mm),
            flipper_length_mm = max(flipper_length_mm))
```

```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = bill_length_mm, y = flipper_length_mm,
             color = species, shape = species)) +
  geom_point(alpha = 0.7) +
  geom_text(data = max_labels, aes(label = island)) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Scatterplot",
       subtitle = "Penguins bill v. flipper length by species",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

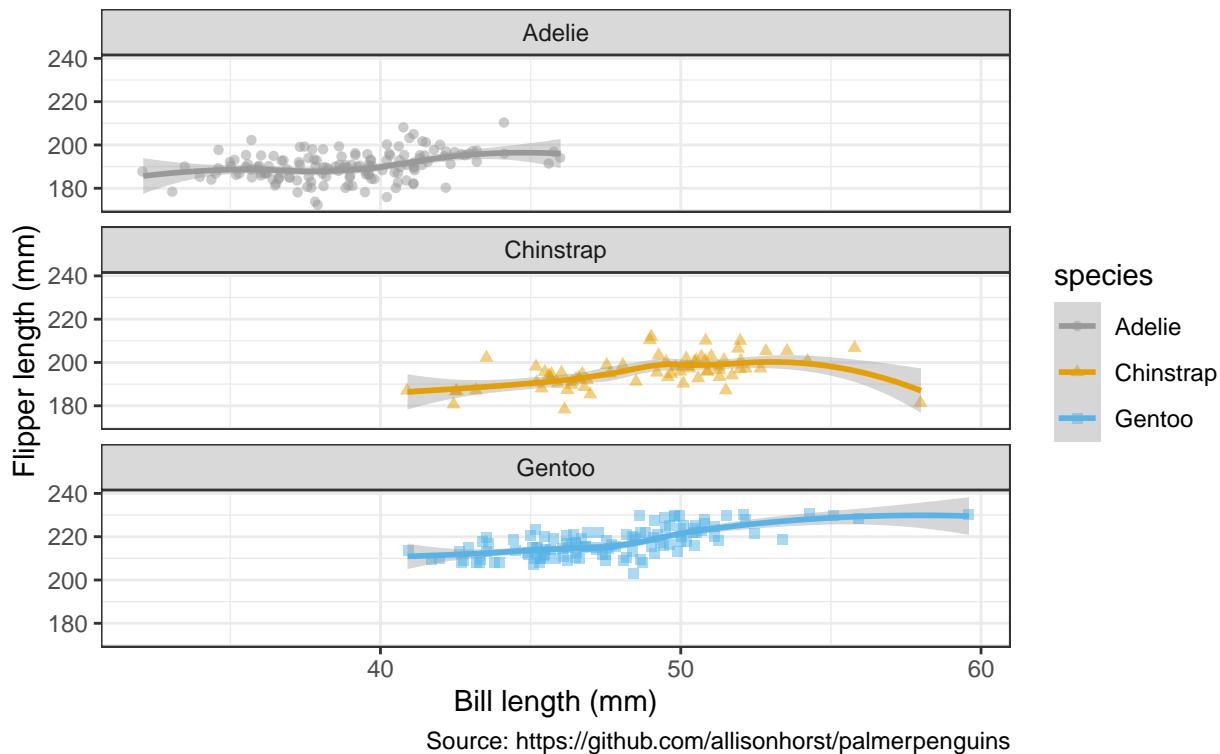


- Jitter with smoothing line

```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = bill_length_mm, y = flipper_length_mm,
             color = species, shape = species)) +
  geom_jitter(alpha = 0.5) +
  geom_smooth(method = "loess", se = TRUE) +
  facet_wrap(vars(species), nrow = 3) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Scatterplot with smoothing line",
       subtitle = "Penguins bill v. flipper length by species with loess smoothing line",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

### Scatterplot with smoothing line

Penguins bill v. flipper length by species with loess smoothing line

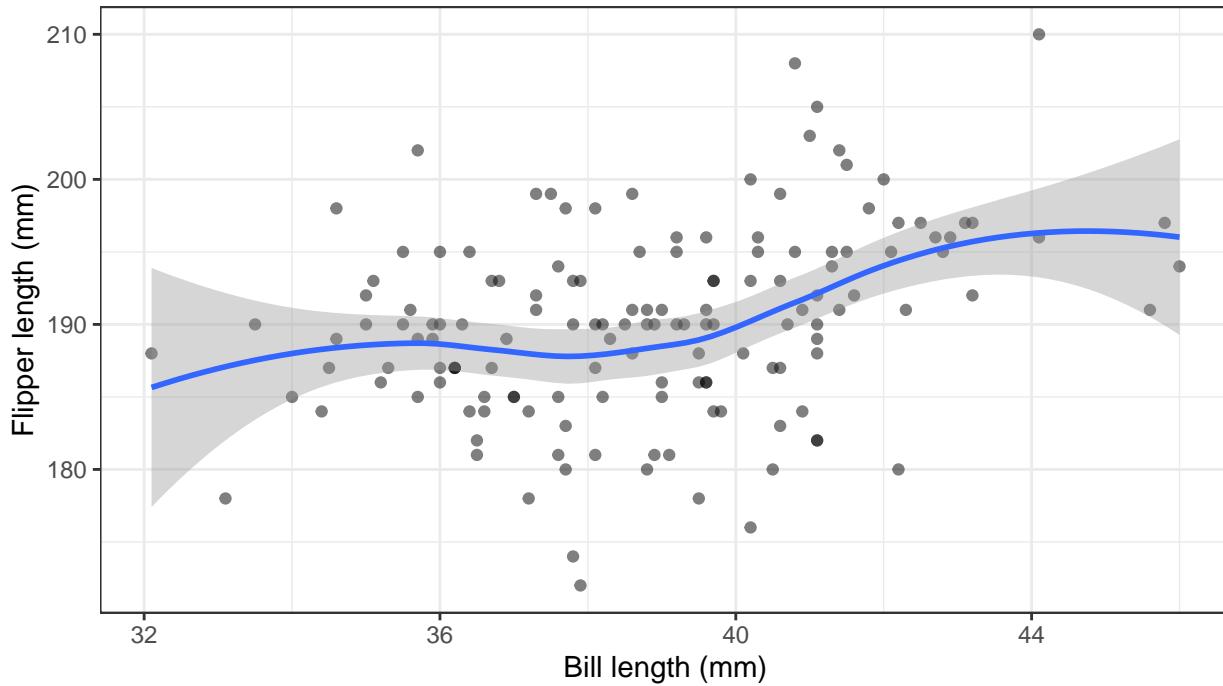


Source: <https://github.com/allisonhorst/palmerpenguins>

```
penguins %>%
  remove_missing() %>%
  filter(species == "Adelie") %>%
  ggplot(aes(x = bill_length_mm, y = flipper_length_mm)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", se = TRUE) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Scatterplot with smoothing line",
       subtitle = "Penguins bill v. flipper length by species with\nloess smoothing line, histogram",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

### Scatterplot with smoothing line

Penguins bill v. flipper length by species with loess smoothing line, histogram & density distribution



Source: <https://github.com/allisonhorst/palmerpenguins>

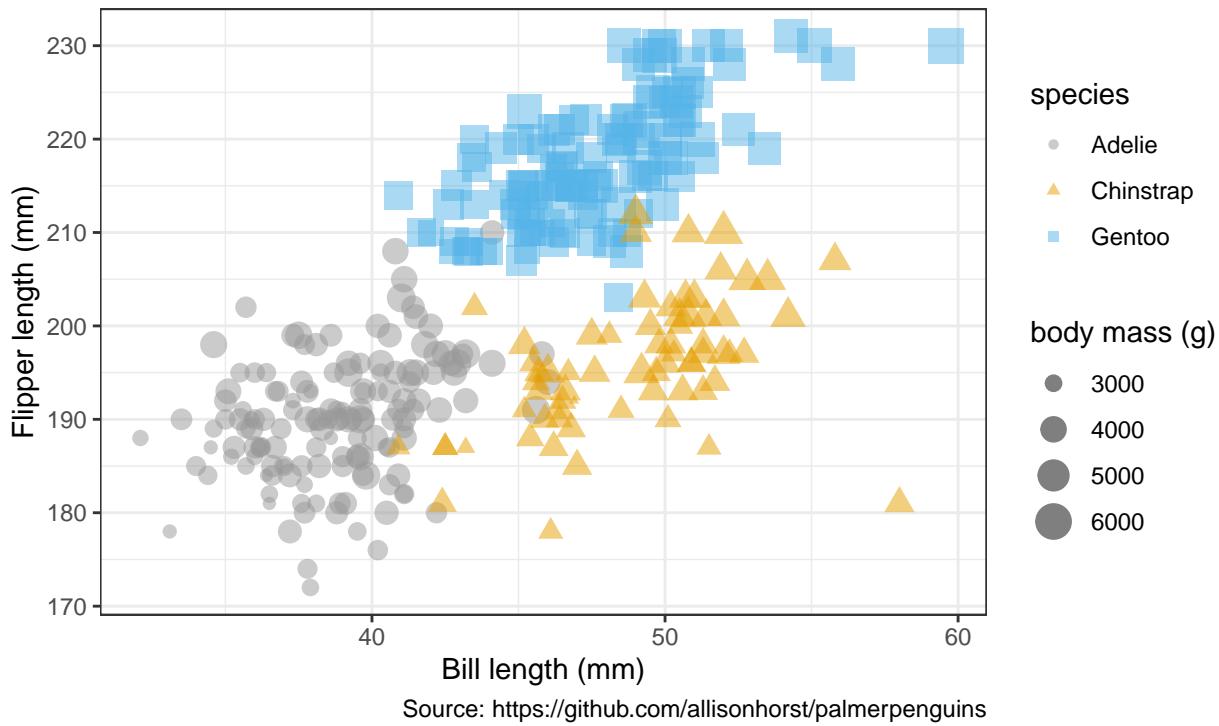
```
#(ggMarginal(p, type = "densigram", fill = "transparent"))
```

## 5.2 气泡图

```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = bill_length_mm, y = flipper_length_mm,
             color = species, shape = species, size = body_mass_g)) +
  geom_point(alpha = 0.5) +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Bubble plot",
       size = "body mass (g)",
       subtitle = "Penguins bill v. flipper length by species;\nsize indicates body mass in grams",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

### Bubble plot

Penguins bill v. flipper length by species;  
size indicates body mass in grams

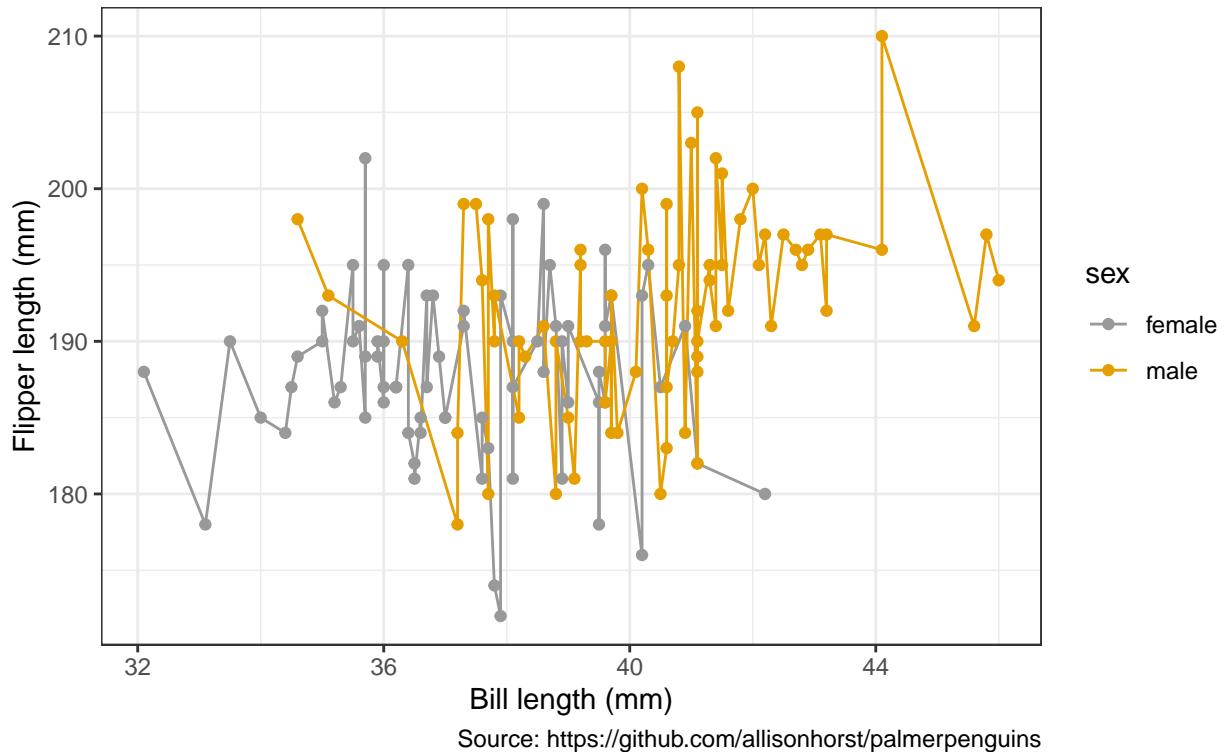


### 5.3 线性图

```
penguins %>%
  remove_missing() %>%
  filter(species == "Adelie") %>%
  ggplot(aes(x = bill_length_mm, y = flipper_length_mm,
             color = sex)) +
  geom_line() +
  geom_point() +
  labs(x = "Bill length (mm)",
       y = "Flipper length (mm)",
       title = "Line plot",
       subtitle = "Penguins bill v. flipper length by species and sex",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

## Line plot

## Penguins bill v. flipper length by species and sex



## 5.4 相关系数图/热力图

```
mat <- penguins %>%
  remove_missing() %>%
  select(bill_depth_mm, bill_length_mm, body_mass_g, flipper_length_mm)

str(mat)

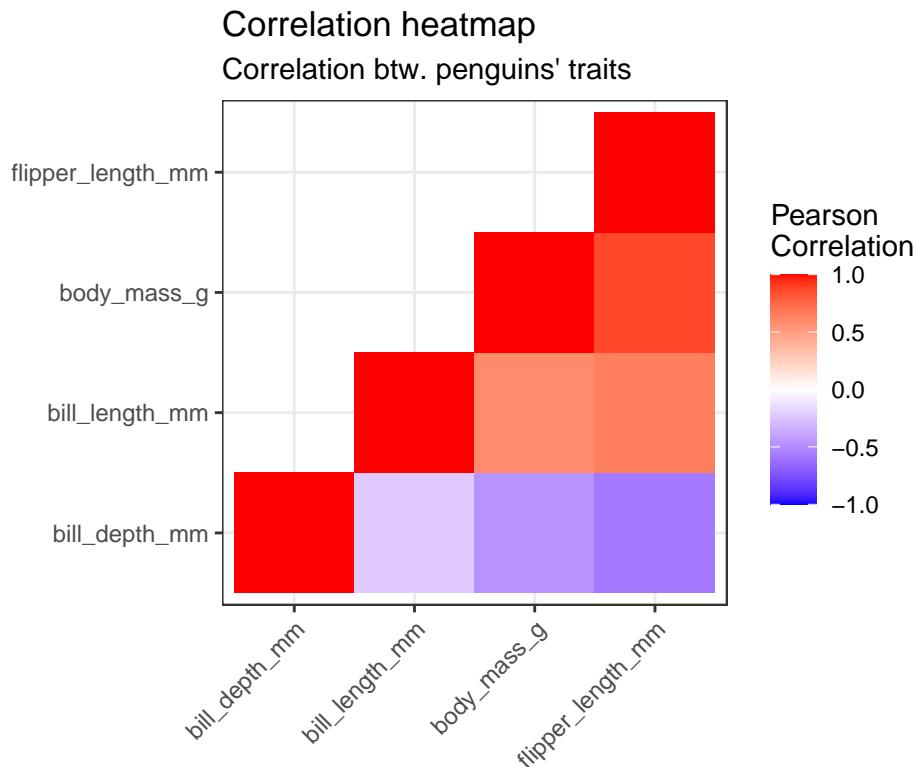
## # tibble [333 x 4] (S3: tbl_df/tbl/data.frame)
## # $ bill_depth_mm      : num [1:333] 18.7 17.4 18 19.3 20.6 ...
## # $ bill_length_mm     : num [1:333] 39.1 39.5 40.3 36.7 39.3 ...
## # $ body_mass_g        : int [1:333] 3750 3800 3250 3450 3650 3625 ...
## # $ flipper_length_mm: int [1:333] 181 186 195 193 190 181 195 182 191 198 ...

cormat <- round(cor(mat), 2)
cormat[upper.tri(cormat)] <- NA

cormat <- cormat %>%
```

```
as_data_frame() %>%
  mutate(x = colnames(mat)) %>%
  gather(key = "y", value = "value", bill_depth_mm:flipper_length_mm)

cormat %>%
  remove_missing() %>%
  arrange(x, y) %>%
  ggplot(aes(x = x, y = y, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
    midpoint = 0, limit = c(-1,1), space = "Lab",
    name = "Pearson\nCorrelation") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  coord_fixed() +
  labs(x = "",
    y = "",
    title = "Correlation heatmap",
    subtitle = "Correlation btw. penguins' traits",
    caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```



Source: <https://github.com/allisonhorst/palmerpenguins>

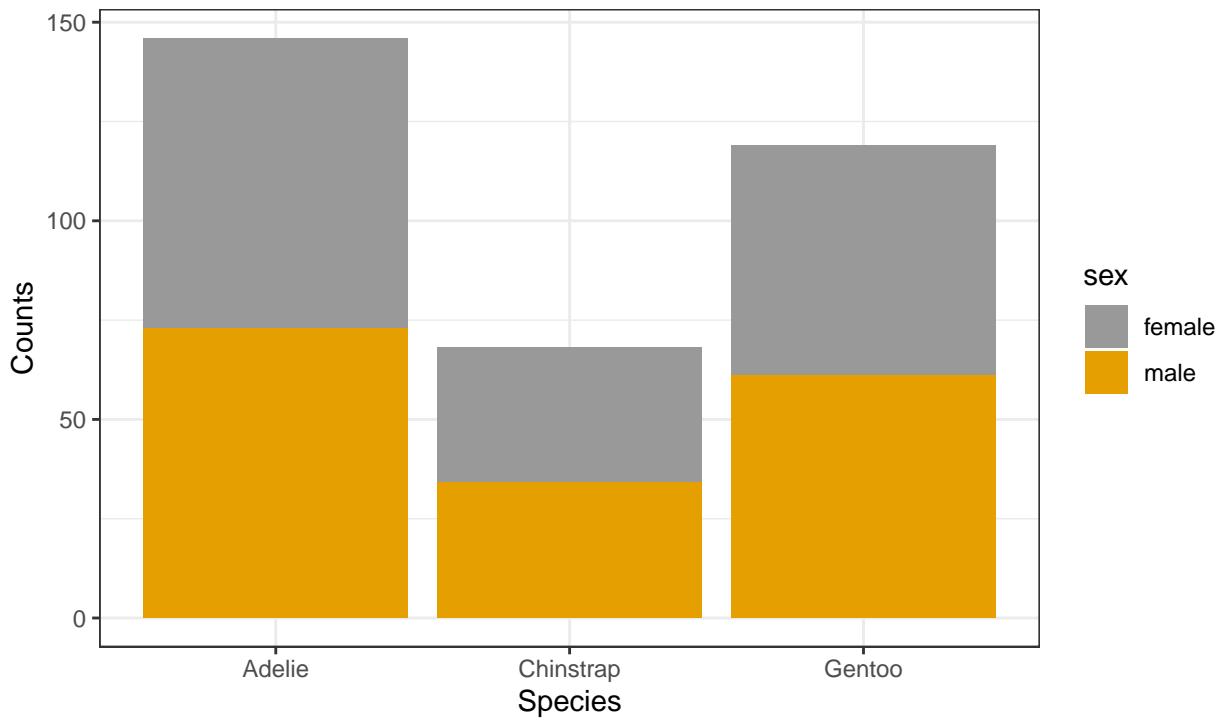
## 5.5 条形图

- per default: counts

```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = species,
             fill = sex)) +
  geom_bar() +
  labs(x = "Species",
       y = "Counts",
       title = "Barchart",
       subtitle = "Counts of male & female penguins per species in study",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

## Barchart

Counts of male & female penguins per species in study

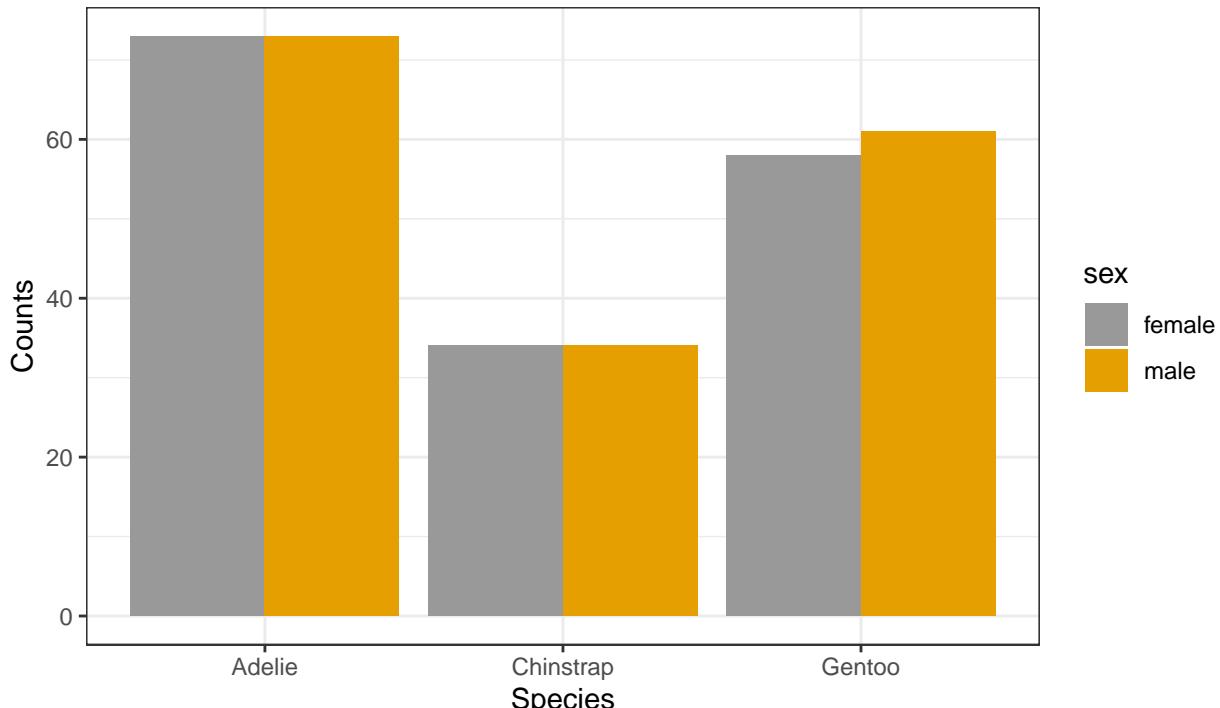


Source: <https://github.com/allisonhorst/palmerpenguins>

```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = species,
             fill = sex)) +
  geom_bar(position = 'dodge') +
  labs(x = "Species",
       y = "Counts",
       title = "Barchart",
       subtitle = "Counts of male & female penguins per species in study",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

## Barchart

Counts of male & female penguins per species in study



Source: <https://github.com/allisonhorst/palmerpenguins>

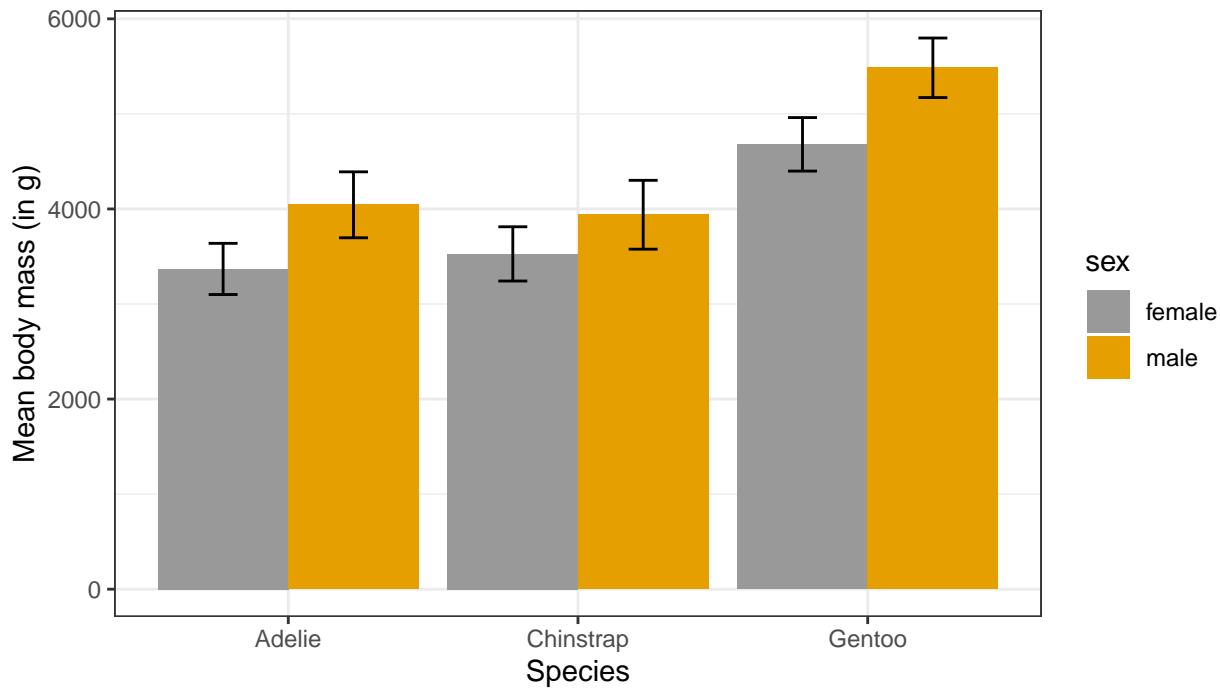
- alternative: set y-values

```
penguins %>%
  remove_missing() %>%
  group_by(species, sex) %>%
  summarise(mean_bmg = mean(body_mass_g),
            sd_bmg = sd(body_mass_g)) %>%
  ggplot(aes(x = species, y = mean_bmg,
             fill = sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = mean_bmg - sd_bmg,
                     ymax = mean_bmg + sd_bmg),
                width = 0.2,
                position = position_dodge(0.9)) +
  labs(x = "Species",
       y = "Mean body mass (in g)",
       title = "Barchart",
       subtitle = "Mean body mass of male & female penguins per species\nwith standard deviation")
```

```
caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

## Barchart

Mean body mass of male & female penguins per species  
with standard deviation



Source: <https://github.com/allisonhorst/palmerpenguins>

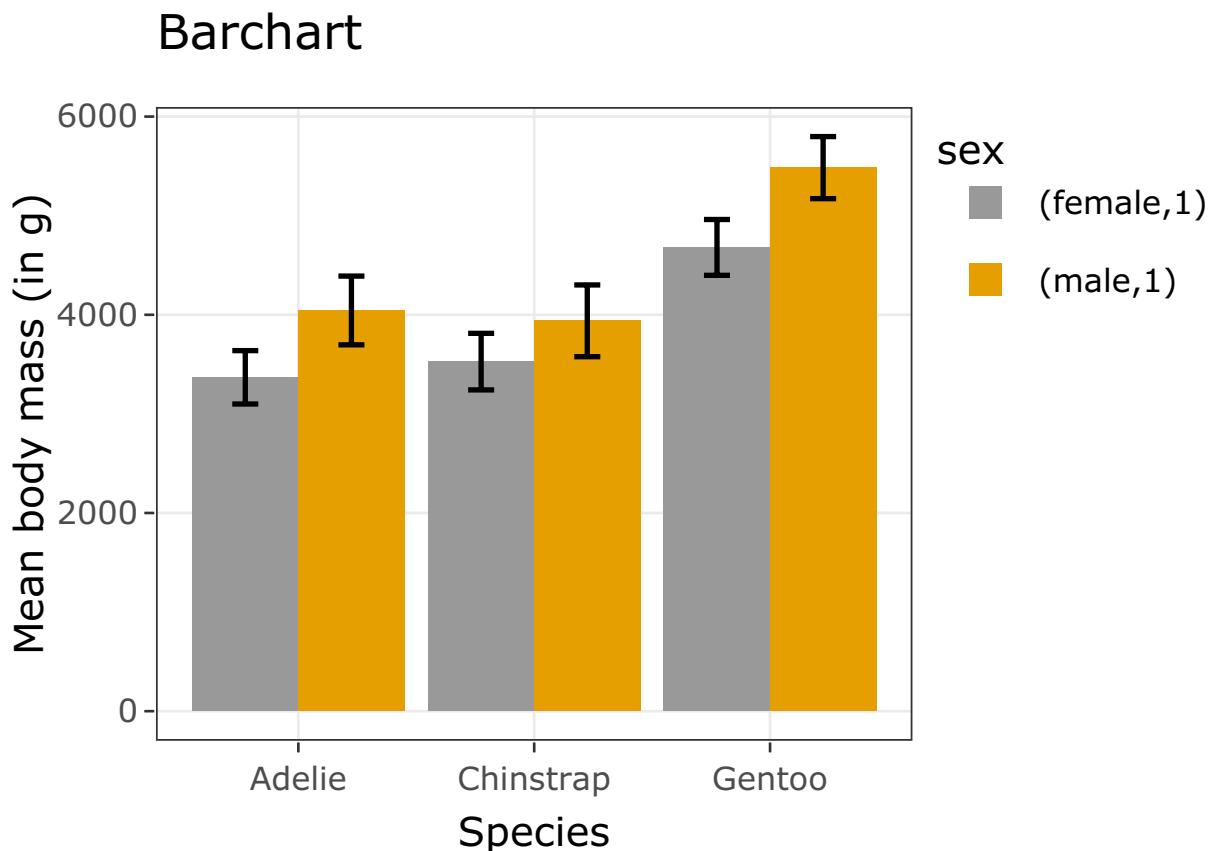
```
library(plotly)

p <- penguins %>%
  remove_missing() %>%
  group_by(species, sex) %>%
  summarise(mean_bmg = mean(body_mass_g),
            sd_bmg = sd(body_mass_g)) %>%
  ggplot(aes(x = species, y = mean_bmg,
             fill = sex)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = mean_bmg - sd_bmg,
                     ymax = mean_bmg + sd_bmg),
                width = 0.2,
                position = position_dodge(0.9)) +
  labs(x = "Species",
       y = "Mean body mass (in g)",
```

```

title = "Barchart",
subtitle = "Mean body mass of male & female penguins per species\nwith standard deviation",
caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

ggplotly(p)

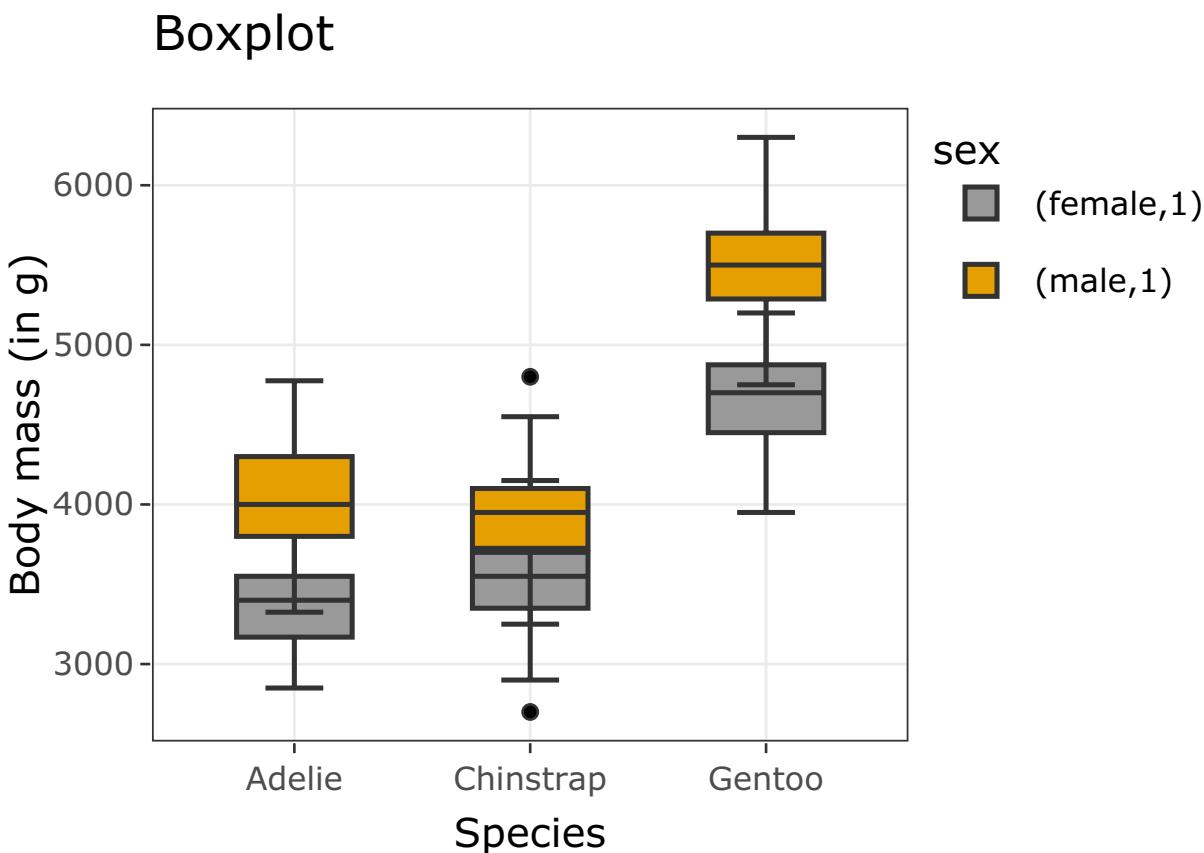


## 5.6 箱线图

```

pp <- penguins %>%
  remove_missing() %>%
  ggplot(aes(x = species, y = body_mass_g,
             fill = sex)) +
  geom_boxplot() +
  labs(x = "Species",
       y = "Body mass (in g)",
       title = "Boxplot",
       subtitle = "Body mass of three penguin species per sex",
```

```
caption = "Source: https://github.com/allisonhorst/palmerpenguins")  
ggplotly(pp)
```

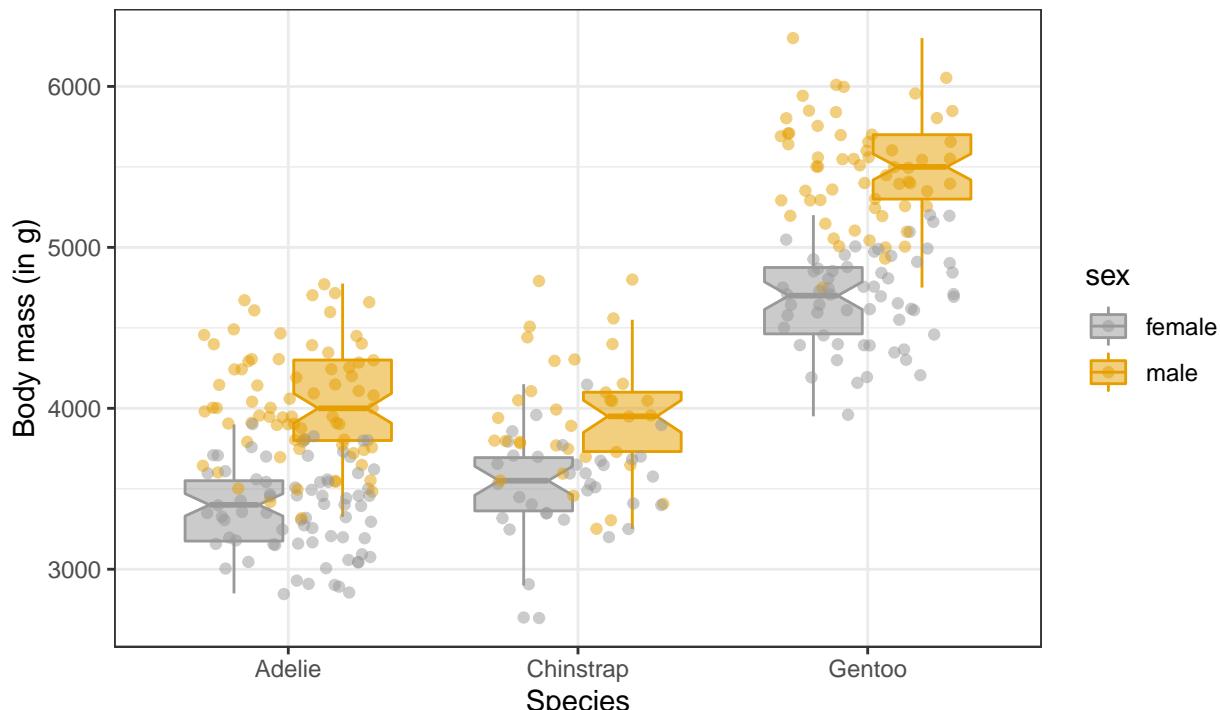


- with points

```
penguins %>%  
  remove_missing() %>%  
  ggplot(aes(x = species, y = body_mass_g,  
             fill = sex, color = sex)) +  
  geom_boxplot(alpha = 0.5, notch = TRUE) +  
  geom_jitter(alpha = 0.5, position=position_jitter(0.3)) +  
  labs(x = "Species",  
       y = "Body mass (in g)",  
       title = "Boxplot with points (dotplot)",  
       subtitle = "Body mass of three penguin species per sex",  
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

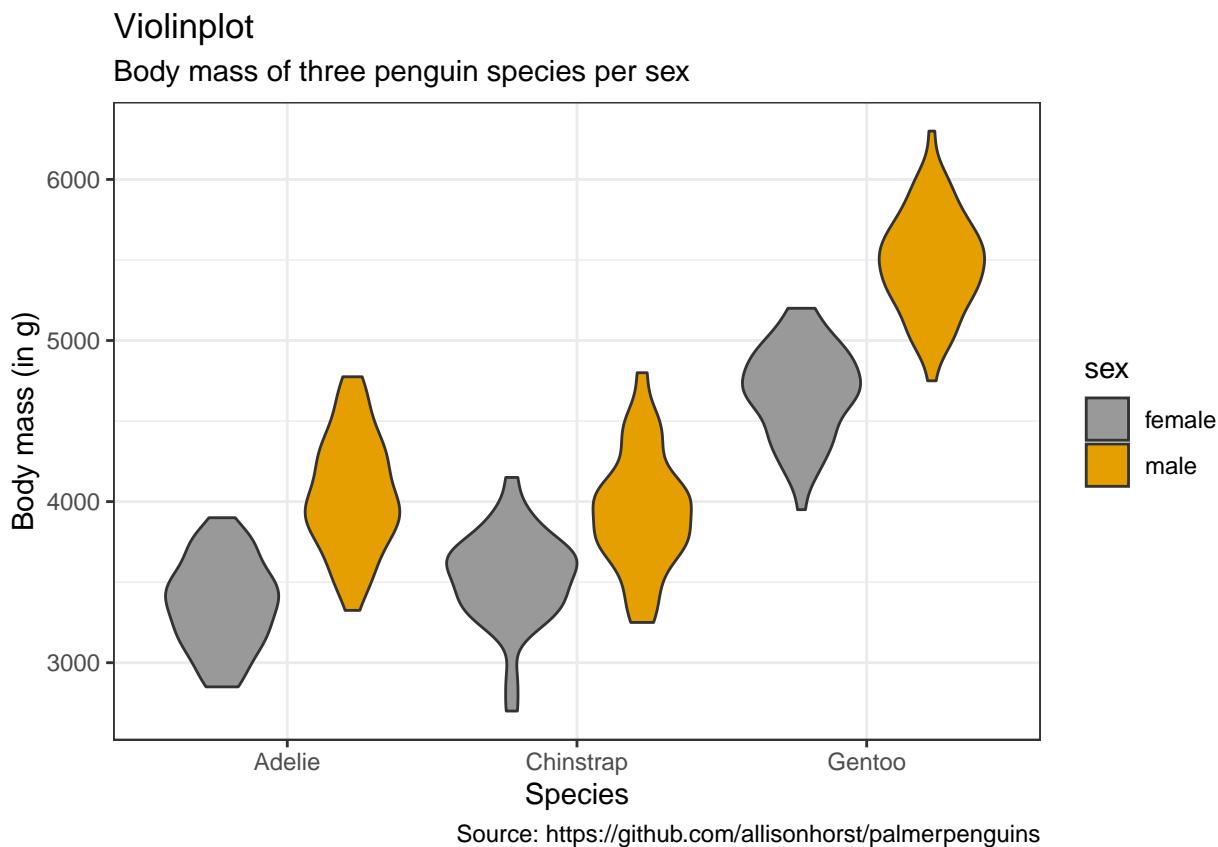
### Boxplot with points (dotplot)

Body mass of three penguin species per sex



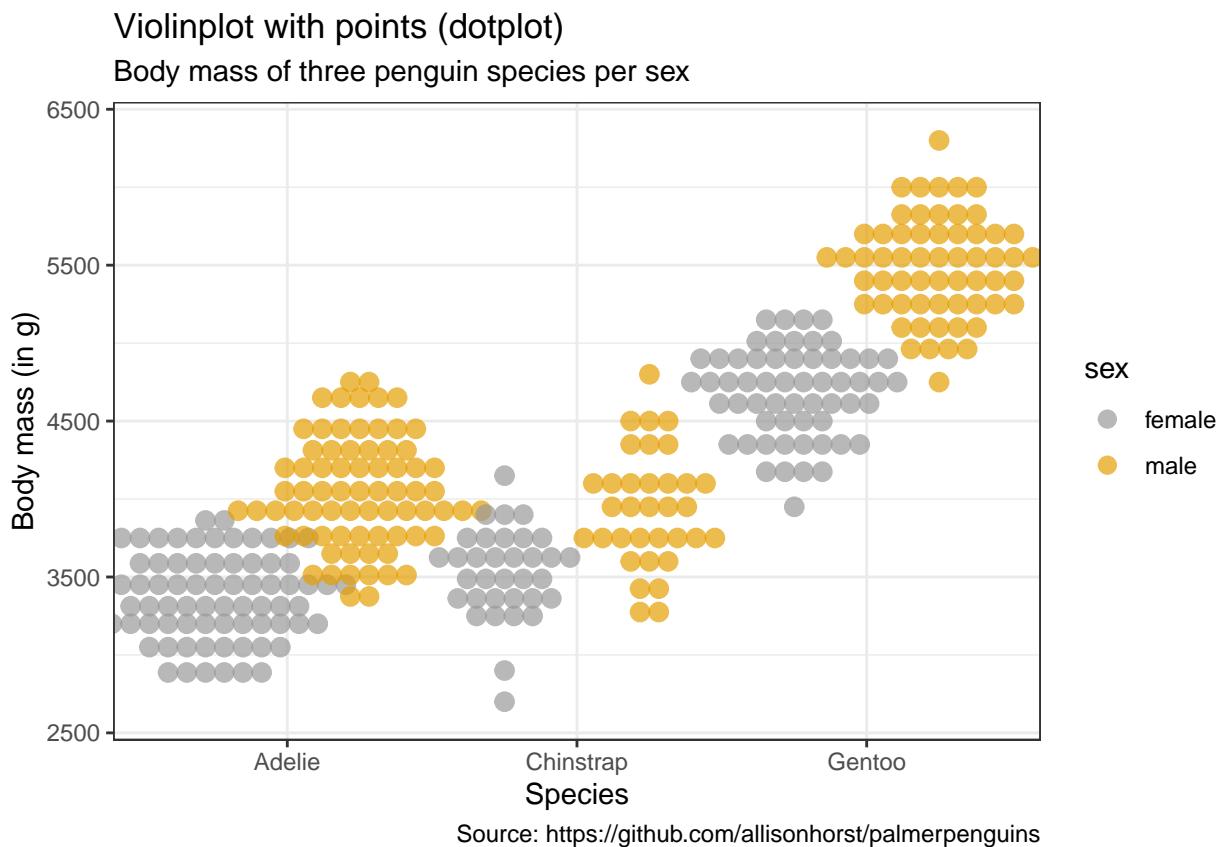
Source: <https://github.com/allisonhorst/palmerpenguins>

```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = species, y = body_mass_g,
             fill = sex)) +
  geom_violin(scale = "area") +
  labs(x = "Species",
       y = "Body mass (in g)",
       title = "Violinplot",
       subtitle = "Body mass of three penguin species per sex",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```



- with dots (sina-plots)

```
penguins %>%
  remove_missing() %>%
  ggplot(aes(x = species, y = body_mass_g,
             fill = sex, color = sex)) +
  geom_dotplot(method = "dotdensity", alpha = 0.7,
               binaxis = 'y', stackdir = 'center',
               position = position_dodge(1)) +
  labs(x = "Species",
       y = "Body mass (in g)",
       title = "Violinplot with points (dotplot)",
       subtitle = "Body mass of three penguin species per sex",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```



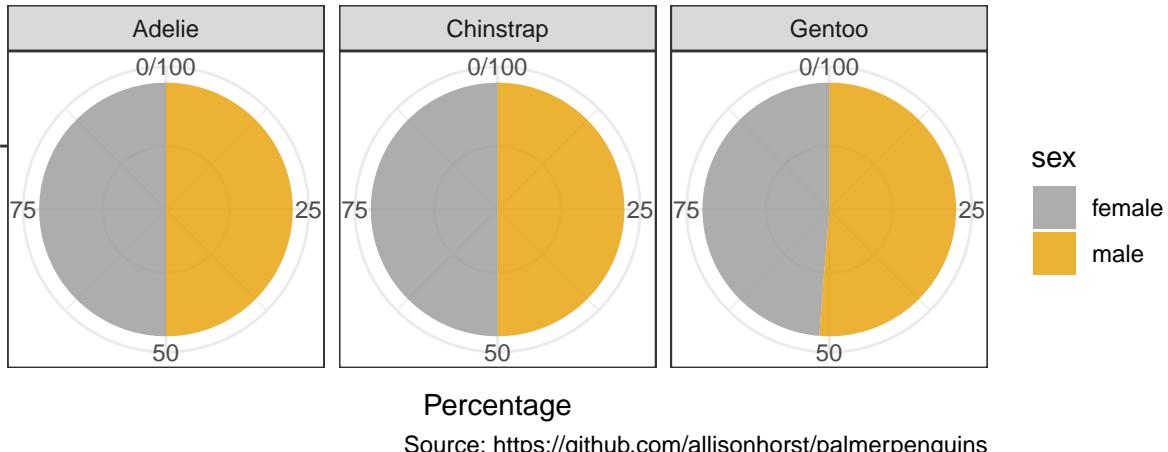
## 5.8 饼图

```
penguins %>%
  remove_missing() %>%
  group_by(species, sex) %>%
  summarise(n = n()) %>%
  mutate(freq = n / sum(n),
        percentage = freq * 100) %>%
  ggplot(aes(x = "", y = percentage,
             fill = sex)) +
  facet_wrap(vars(species), nrow = 1) +
  geom_bar(stat = "identity", alpha = 0.8) +
  coord_polar("y") +
  labs(x = "",
       y = "Percentage",
       title = "Piechart",
       subtitle = "Percentage of male v. female penguins per species in study",
```

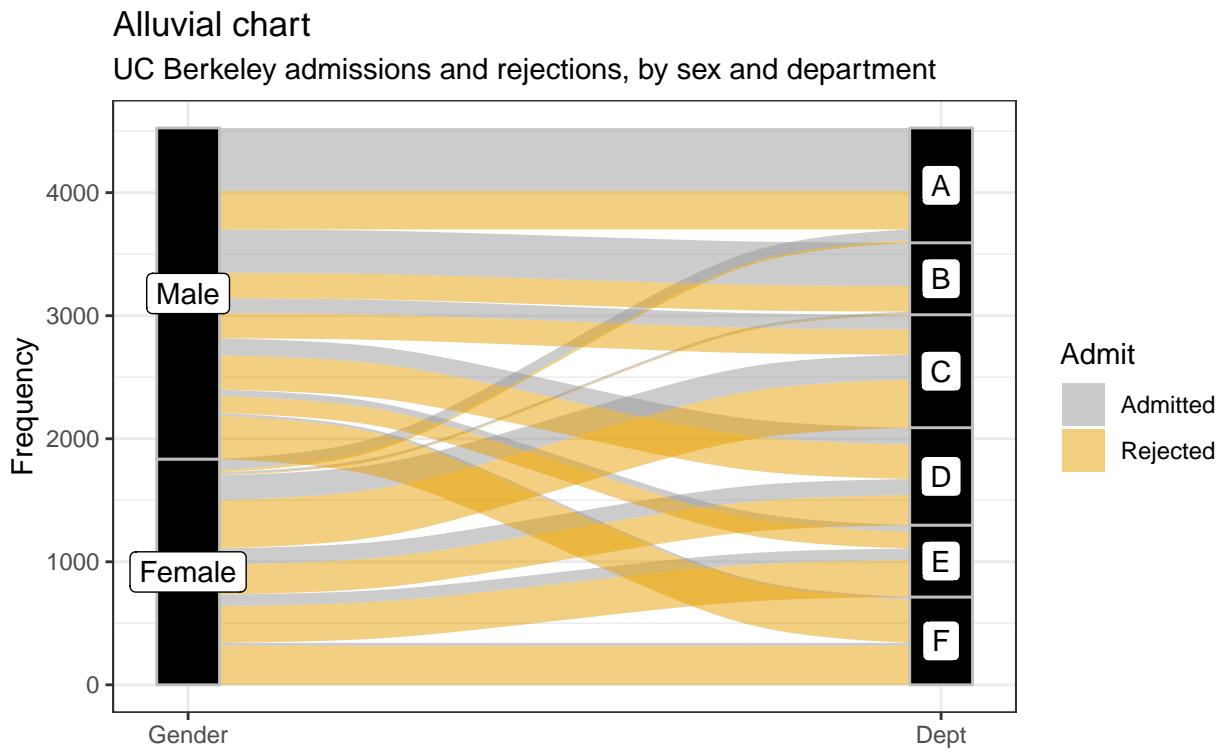
```
caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

## Piechart

Percentage of male v. female penguins per species in study



```
as.data.frame(UCBAdmissions) %>%
  ggplot(aes(y = Freq, axis1 = Gender, axis2 = Dept)) +
  geom_alluvium(aes(fill = Admit), width = 1/12) +
  geom_stratum(width = 1/12, fill = "black", color = "grey") +
  geom_label(stat = "stratum", aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("Gender", "Dept"), expand = c(.05, .05)) +
  labs(x = "",
       y = "Frequency",
       title = "Alluvial chart",
       subtitle = "UC Berkeley admissions and rejections, by sex and department",
       caption = "Source: Bickel et al. (1975)\nSex bias in graduate admissions: Data from Berkeley")
```



Source: Bickel et al. (1975)  
Sex bias in graduate admissions: Data from Berkeley. Science, 187, 398...403.

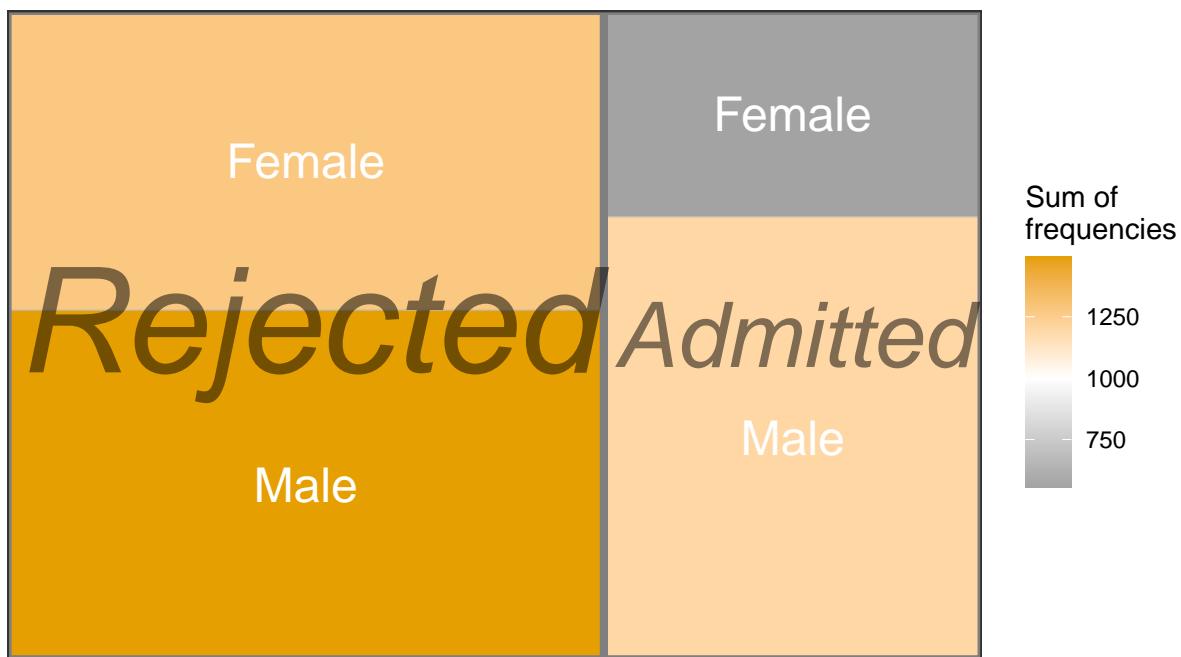
## 5.10 系谱图

```
as.data.frame(UCBAmissions) %>%
  group_by(Admit, Gender) %>%
  summarise(sum_freq = sum(Freq)) %>%
  ggplot(aes(area = sum_freq, fill = sum_freq, label = Gender,
             subgroup = Admit)) +
  geom_treemap() +
  geom_treemap_subgroup_border() +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha = 0.5, colour =
    "black", fontface = "italic", min.size = 0) +
  geom_treemap_text(colour = "white", place = "centre", reflow = T) +
  scale_fill_gradient2(low = "#999999", high = "#E69F00", mid = "white", midpoint = 1000, space =
    "name = "Sum of\nfrequencies") +
  labs(x = "",
       y = "",
       title = "Treemap",
```

```
subtitle = "UC Berkeley admissions and rejections by sex",
caption = "Source: Bickel et al. (1975)\nSex bias in graduate admissions: Data from Berkeley"
```

### Treemap

UC Berkeley admissions and rejections by sex



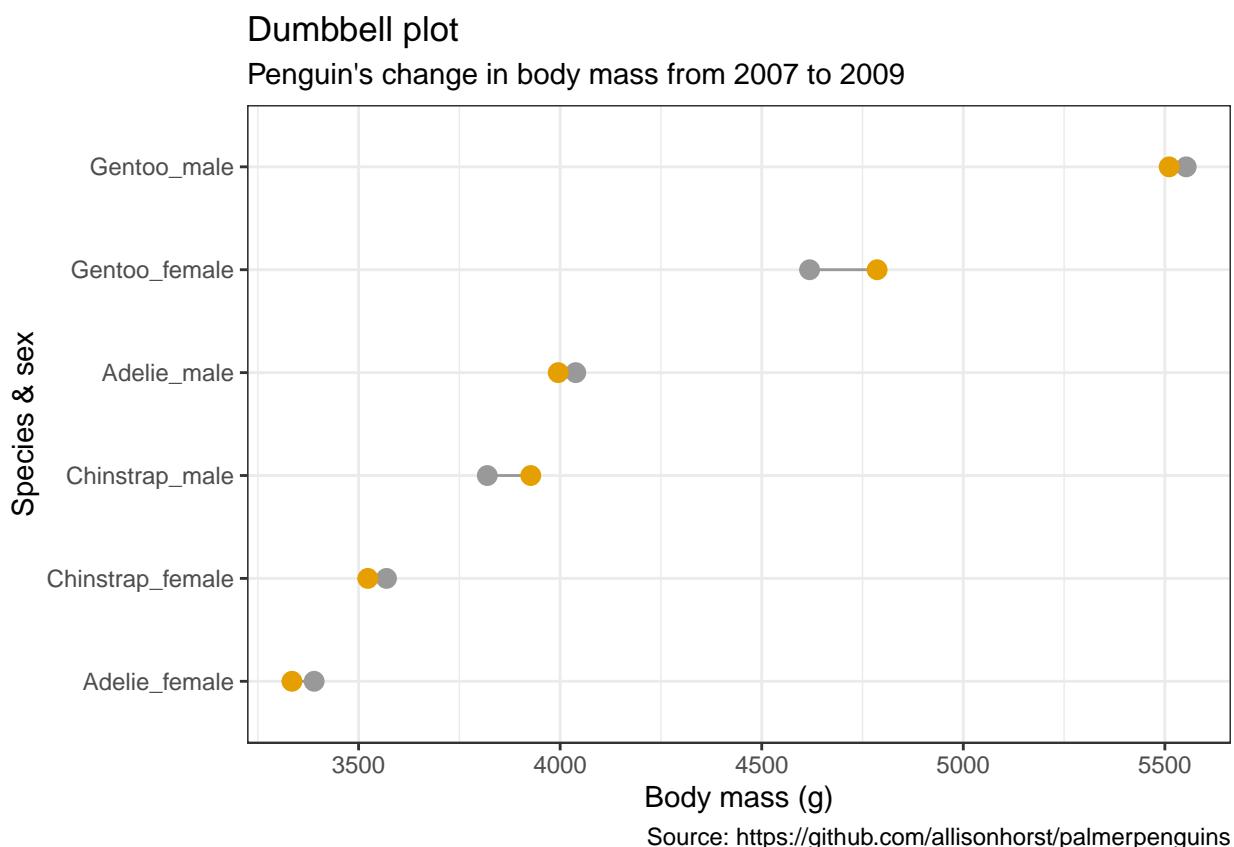
Source: Bickel et al. (1975)  
Sex bias in graduate admissions: Data from Berkeley. Science, 187, 398...403.

## 5.11 哑铃图

```
penguins %>%
  remove_missing() %>%
  group_by(year, species, sex) %>%
  summarise(mean_bmg = mean(body_mass_g)) %>%
  mutate(species_sex = paste(species, sex, sep = "_"),
         year = paste0("year_", year)) %>%
  spread(year, mean_bmg) %>%
  ggplot(aes(x = year_2007, xend = year_2009,
             y = reorder(species_sex, year_2009))) +
  geom_dumbbell(color = "#999999",
                size_x = 3,
                size_xend = 3,
```

```
#Note: there is no US:'color' for UK:'colour'
# in geom_dumbbell unlike standard geoms in ggplot()
colour_x = "#999999",
colour_xend = "#E69F00") +
  labs(x = "Body mass (g)",
       y = "Species & sex",
       title = "Dumbbell plot",
       subtitle = "Penguin's change in body mass from 2007 to 2009",
       caption = "Source: https://github.com/allisonhorst/palmerpenguins")

```



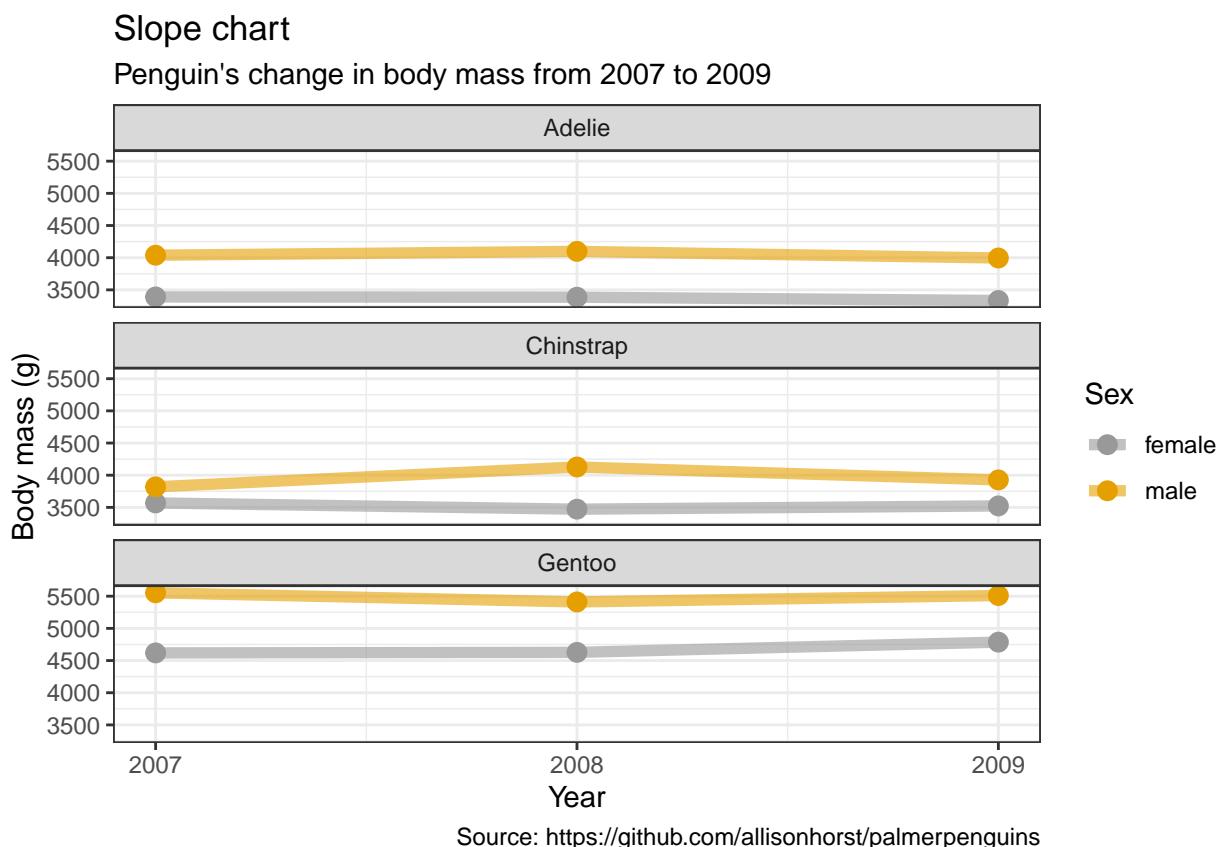
## 5.12 斜率图

```
penguins %>%
  remove_missing() %>%
  group_by(year, species, sex) %>%
  summarise(mean_bmg = mean(body_mass_g)) %>%
  ggplot(aes(x = year, y = mean_bmg, group = sex,
```

```

            color = sex) +
facet_wrap(vars(species), nrow = 3) +
geom_line(alpha = 0.6, size = 2) +
geom_point(alpha = 1, size = 3) +
scale_x_continuous(breaks=c(2007, 2008, 2009)) +
labs(x = "Year",
y = "Body mass (g)",
color = "Sex",
title = "Slope chart",
subtitle = "Penguin's change in body mass from 2007 to 2009",
caption = "Source: https://github.com/allisonhorst/palmerpenguins")

```



## 5.13 堆叠面积图

```

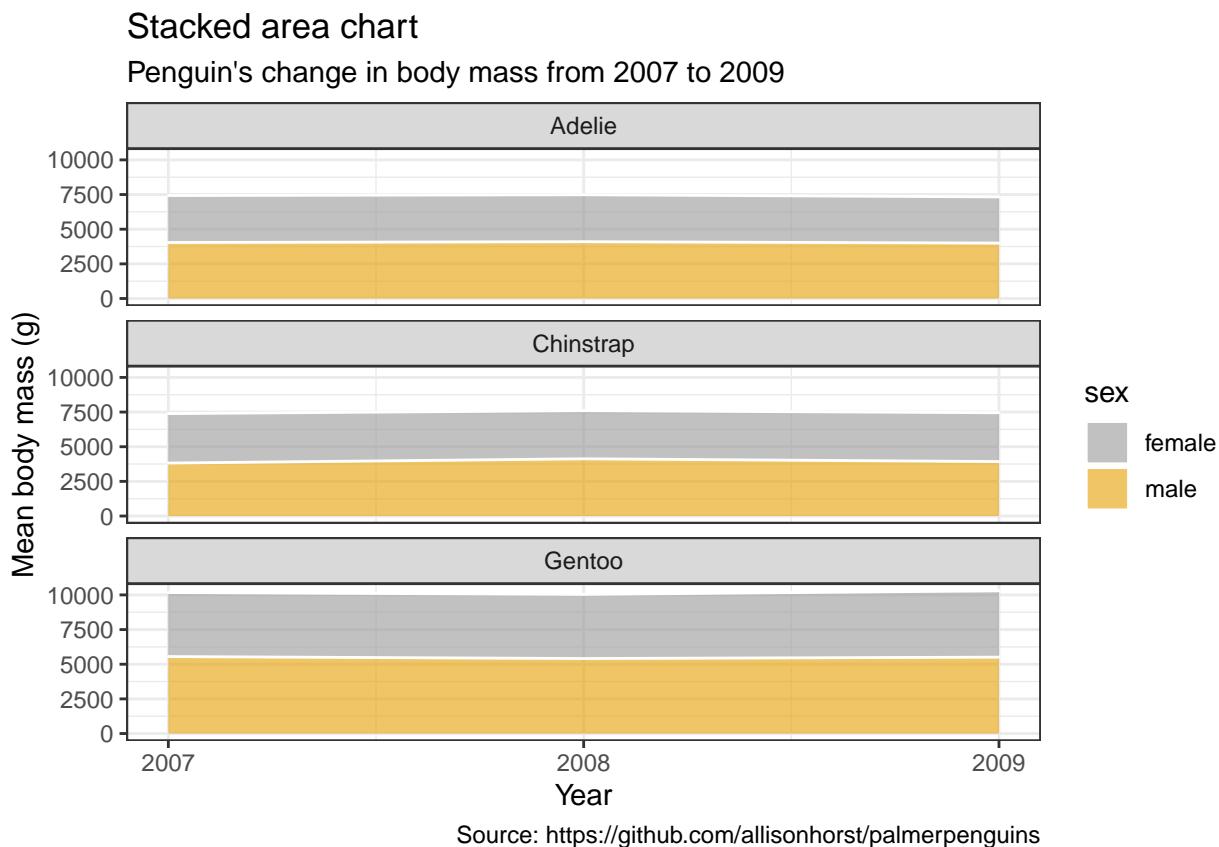
penguins %>%
remove_missing() %>%
group_by(year, species, sex) %>%

```

```

summarise(mean_bmg = mean(body_mass_g)) %>%
ggplot(aes(x = year, y = mean_bmg, fill = sex)) +
facet_wrap(vars(species), nrow = 3) +
geom_area(alpha = 0.6, size=.5, color = "white") +
scale_x_continuous(breaks=c(2007, 2008, 2009)) +
labs(x = "Year",
y = "Mean body mass (g)",
color = "Sex",
title = "Stacked area chart",
subtitle = "Penguin's change in body mass from 2007 to 2009",
caption = "Source: https://github.com/allisonhorst/palmerpenguins")

```



## 5.14 棒棒糖图

```

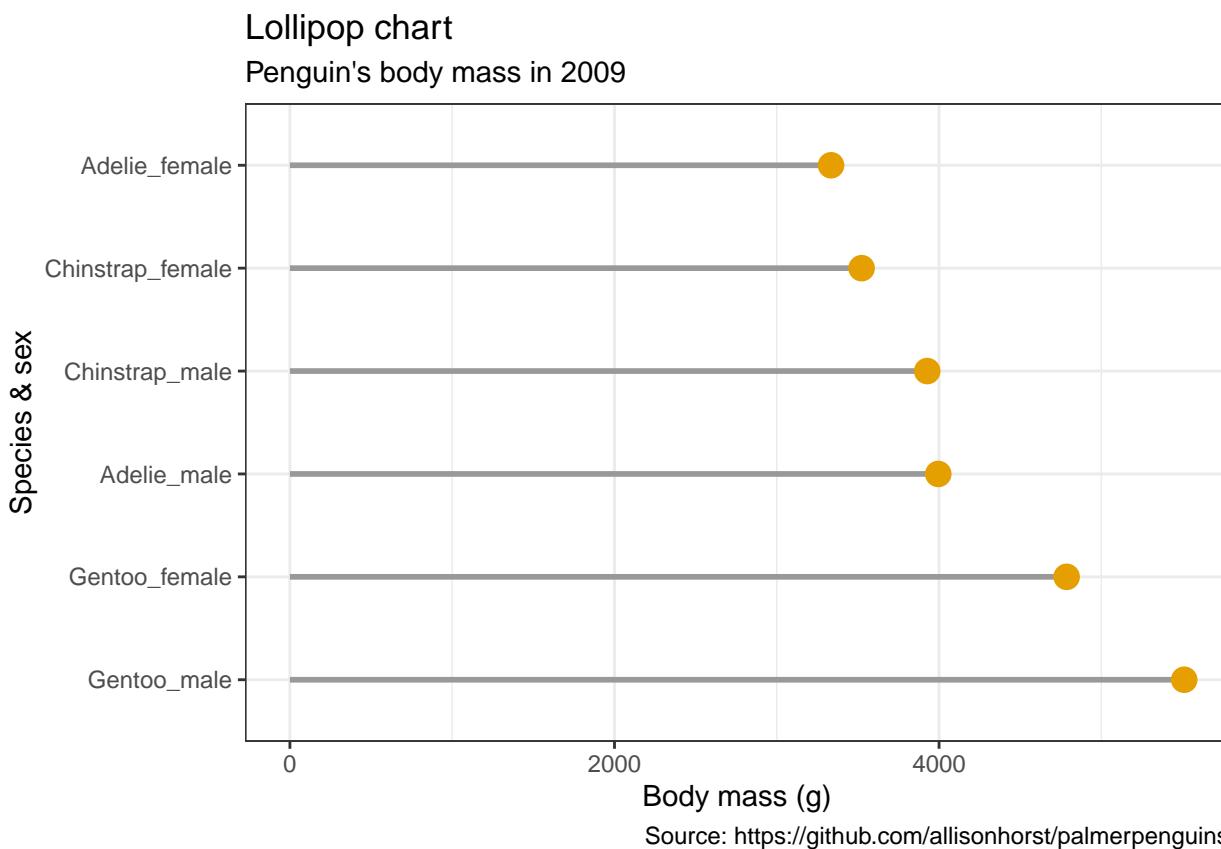
penguins %>%
remove_missing() %>%
group_by(year, species, sex) %>%

```

```

summarise(mean_bmg = mean(body_mass_g)) %>%
mutate(species_sex = paste(species, sex, sep = " "),
       year = paste0("year_", year)) %>%
spread(year, mean_bmg) %>%
ggplot() +
geom_segment(aes(x = reorder(species_sex, -year_2009), xend = reorder(species_sex, -year_2009),
                  y = 0, yend = year_2009),
              color = "#999999", size = 1) +
geom_point(aes(x = reorder(species_sex, -year_2009), y = year_2009),
            size = 4, color = "#E69F00") +
coord_flip() +
labs(x = "Species & sex",
     y = "Body mass (g)",
     title = "Lollipop chart",
     subtitle = "Penguin's body mass in 2009",
     caption = "Source: https://github.com/allisonhorst/palmerpenguins")

```



## 5.15 树状图

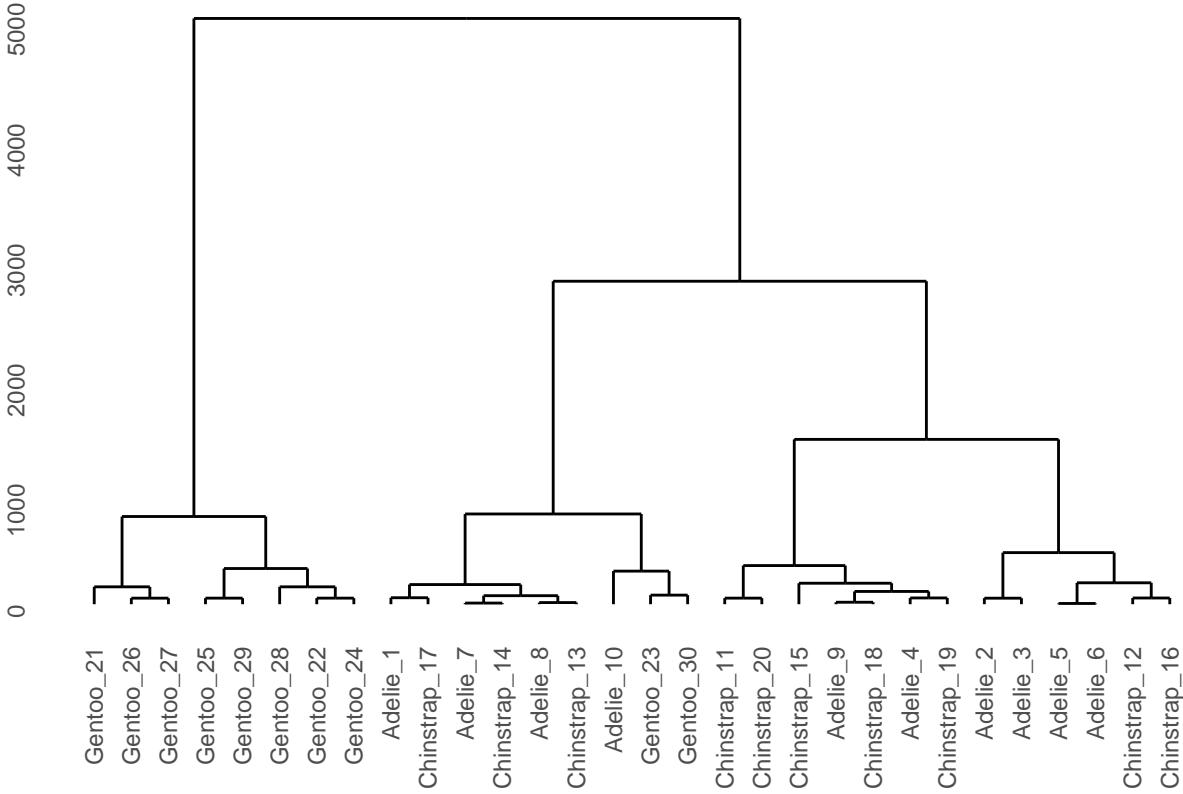
```
library(ggdendro)
library(dendextend)

penguins_hist <- penguins %>%
  filter(sex == "male") %>%
  select(species, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g) %>%
  group_by(species) %>%
  sample_n(10) %>%
  as.data.frame()

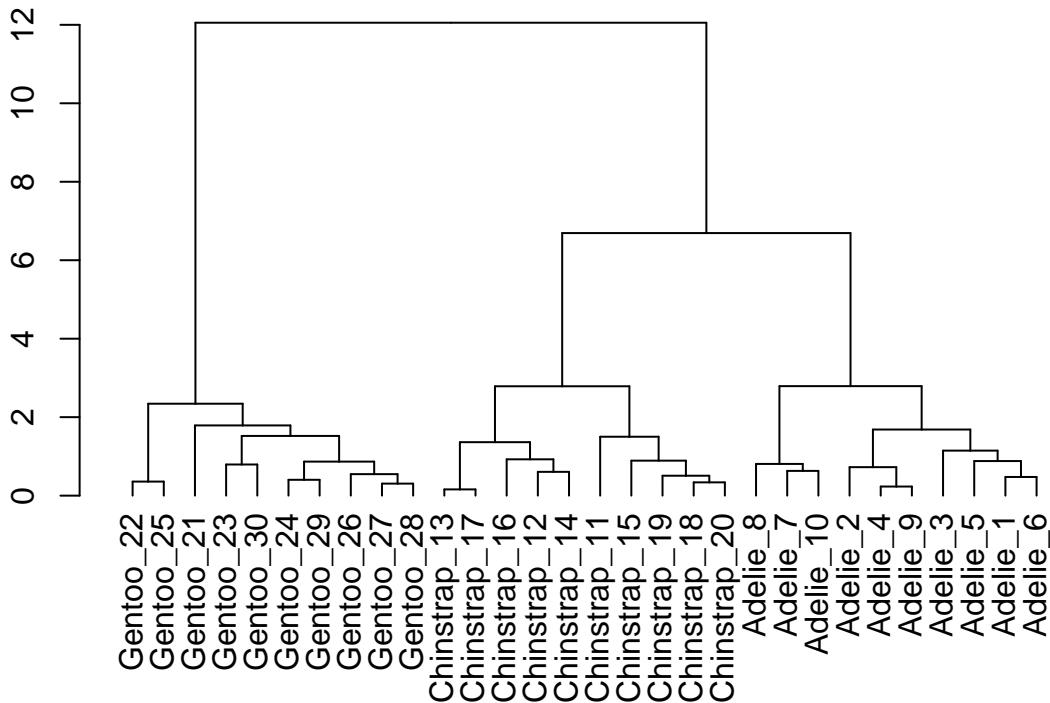
rownames(penguins_hist) <- paste(penguins_hist$species, seq_len(nrow(penguins_hist)), sep = "_")

penguins_hist <- penguins_hist %>%
  select(-species) %>%
  remove_missing()

hc <- hclust(dist(penguins_hist), method = "euclidean"), method = "ward.D2")
ggdendrogram(hc)
```



```
# Create a dendrogram and plot it
penguins_hist %>%
  scale %>%
  dist(method = "euclidean") %>%
  hclust(method = "ward.D2") %>%
  as.dendrogram() %>%
  plot()
```



## 5.16 瀑布图

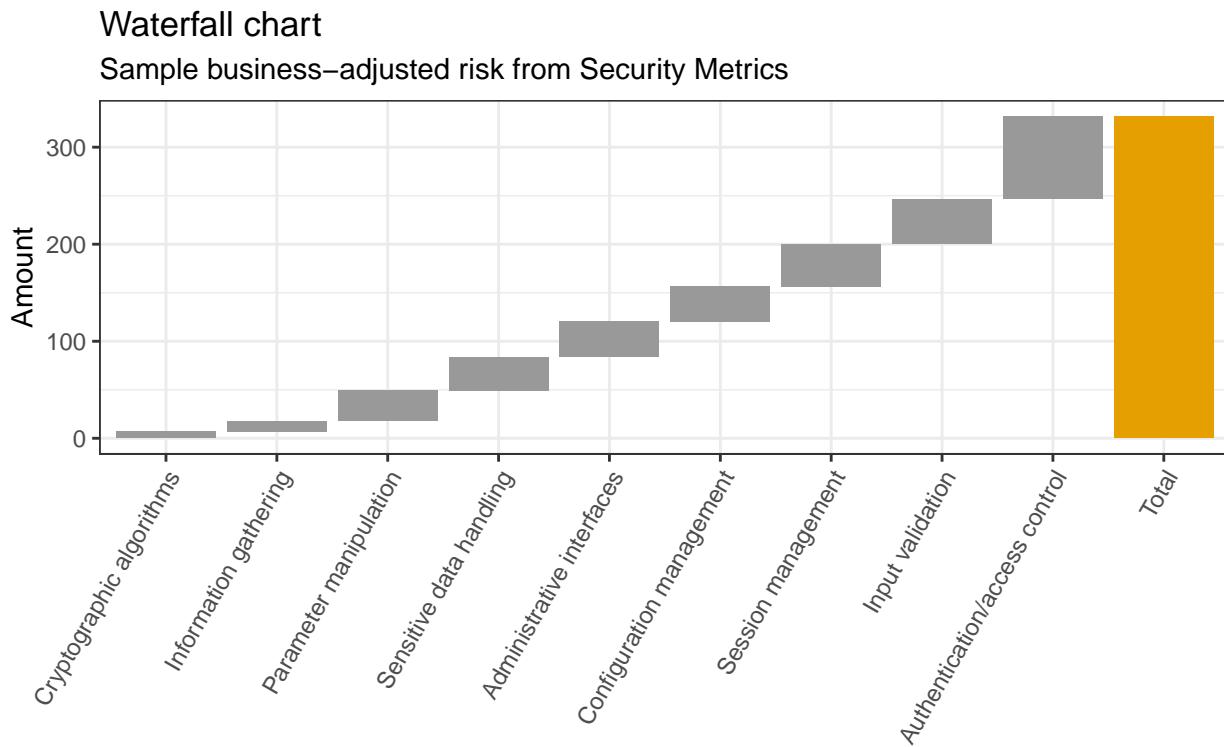
```
library(waterfall)

jaquith %>%
  arrange(score) %>%
  add_row(factor = "Total", score = sum(jaquith$score)) %>%
  mutate(factor = factor(factor, levels = factor),
        id = seq_along(score)) %>%
  mutate(end = cumsum(score),
        start = c(0, end[-length(end)]),
        start = c(start[-length(start)], 0),
        end = c(end[-length(end)], score[length(score)]),
```

```

gr_col = ifelse(factor == "Total", "Total", "Part")) %>%
ggplot(aes(x = factor, fill = gr_col)) +
  geom_rect(aes(x = factor,
                 xmin = id - 0.45, xmax = id + 0.45,
                 ymin = end, ymax = start)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 1, hjust = 1),
        legend.position = "none") +
  labs(x = "",
       y = "Amount",
       title = "Waterfall chart",
       subtitle = "Sample business-adjusted risk from Security Metrics",
       caption = "Andrew Jaquith, Security Metrics: Replacing Fear, Uncertainty, and Doubt\n(Boston: Addison-Wesley Professional, 2007), 170–171")

```



Andrew Jaquith, Security Metrics: Replacing Fear, Uncertainty, and Doubt  
(Boston: Addison-Wesley Professional, 2007), 170–171.

## 5.17 双标图

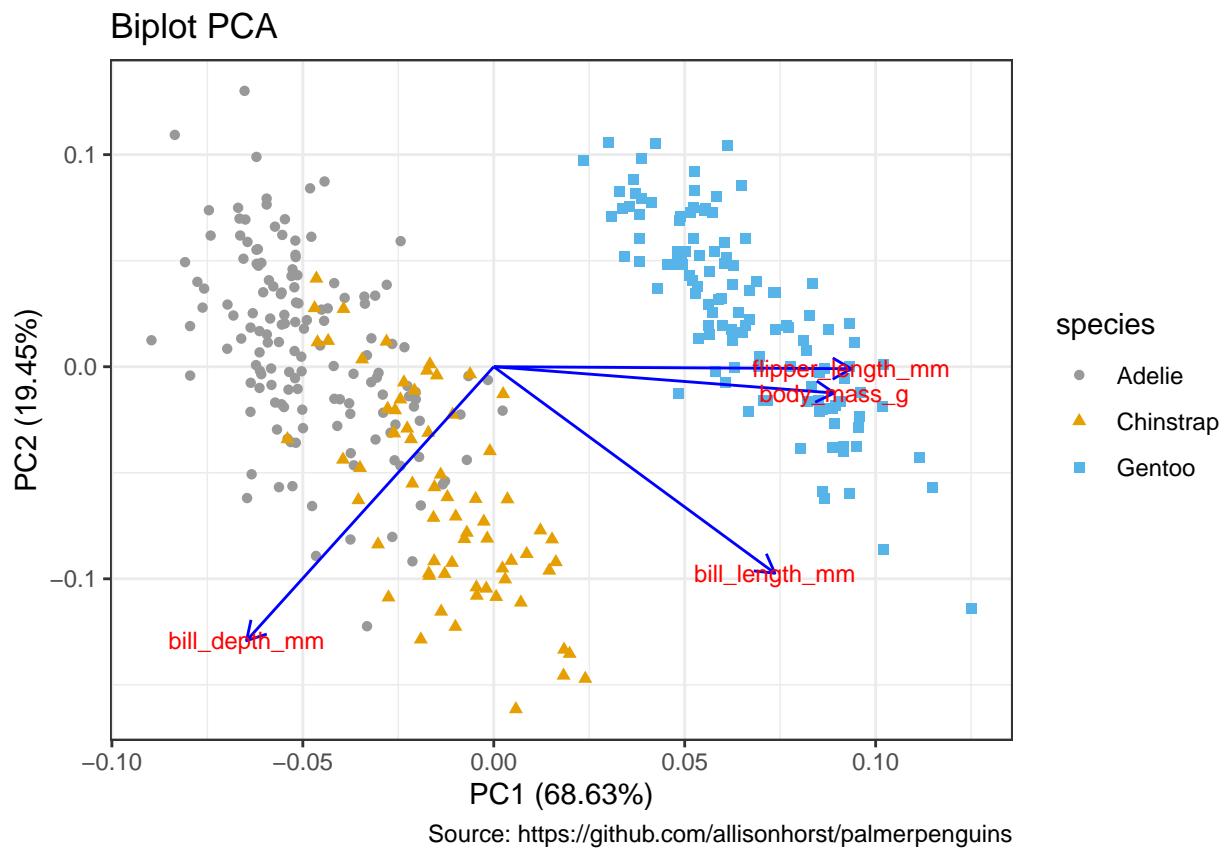
```
library(ggfortify)
```

```
penguins_prep <- penguins %>%
  remove_missing() %>%
  select(bill_length_mm:body_mass_g)

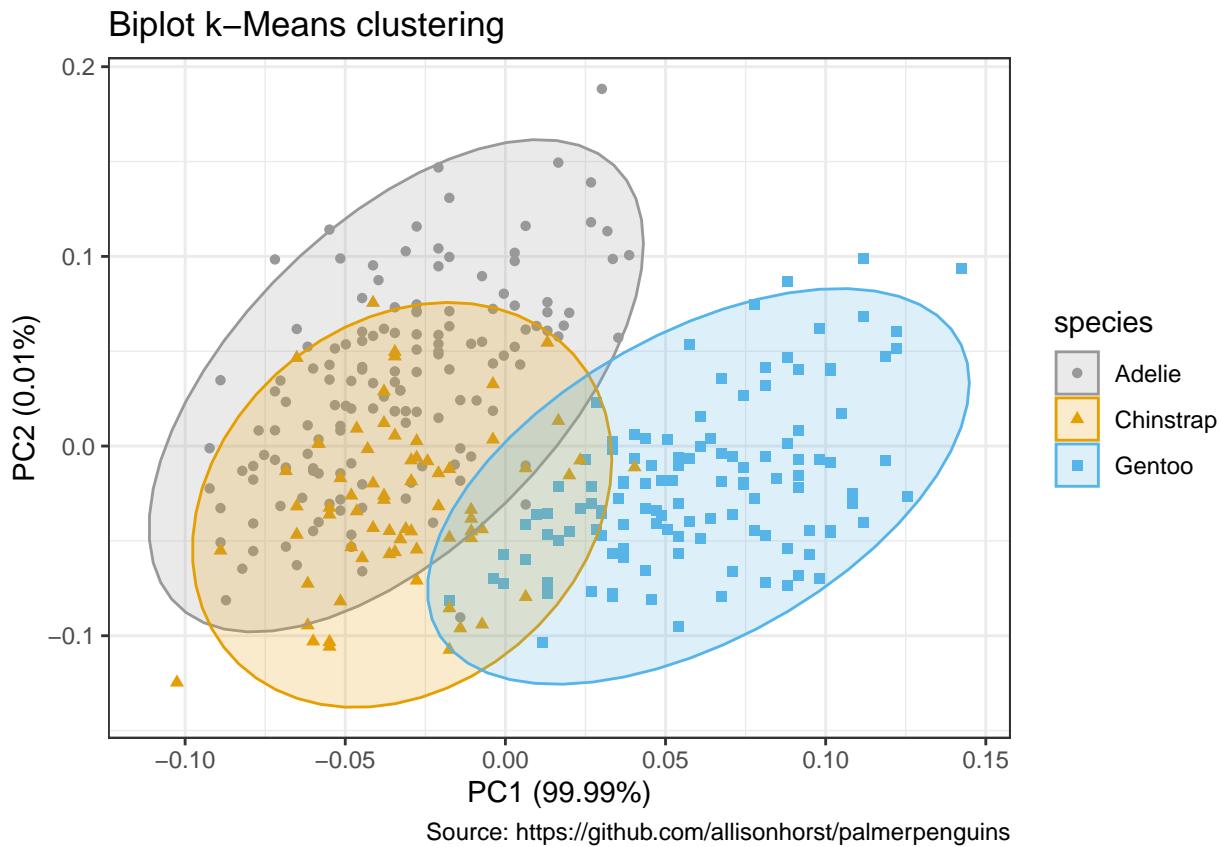
penguins_pca <- penguins_prep %>%
  prcomp(scale. = TRUE)

penguins_km <- penguins_prep %>%
  kmeans(3)

autoplot(penguins_pca,
         data = penguins %>% remove_missing(),
         colour = 'species',
         shape = 'species',
         loadings = TRUE,
         loadings.colour = 'blue',
         loadings.label = TRUE,
         loadings.label.size = 3) +
  scale_color_manual(values = cbp1) +
  scale_fill_manual(values = cbp1) +
  theme_bw() +
  labs(
    title = "Biplot PCA",
    caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```



```
autoplot(penguins_km,
         data = penguins %>% remove_missing(),
         colour = 'species',
         shape = 'species',
         frame = TRUE, frame.type = 'norm') +
  scale_color_manual(values = cbp1) +
  scale_fill_manual(values = cbp1) +
  theme_bw() +
  labs(
    title = "Biplot k-Means clustering",
    caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```



## 5.18 雷达图/星图/蜘蛛图

<https://www.data-to-viz.com/caveat/spider.html>

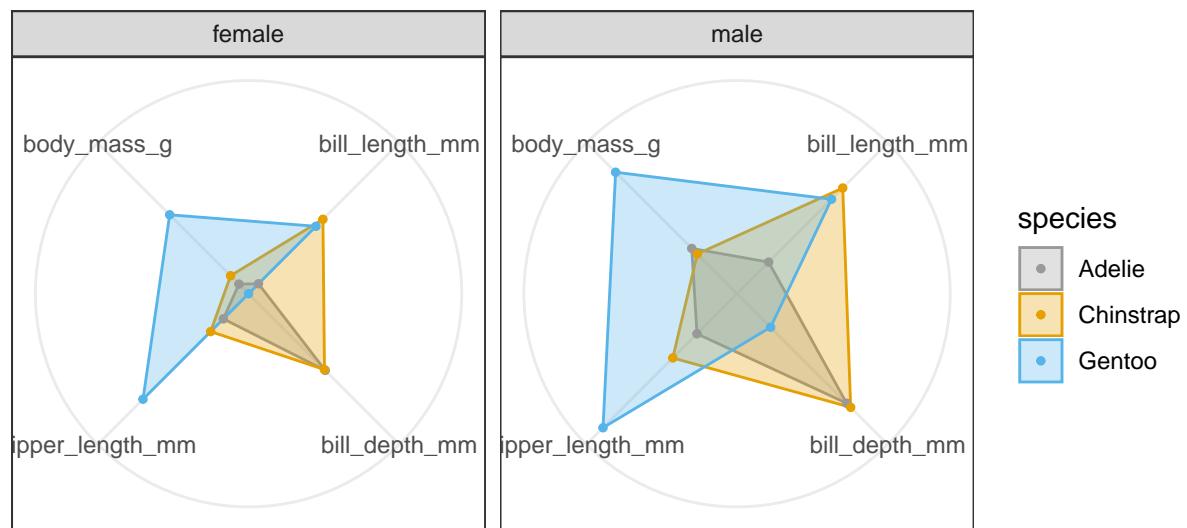
```
library(ggiraphExtra)
```

```
penguins %>%
  remove_missing() %>%
  select(-island, -year) %>%
  ggRadar(aes(x = c(bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g),
    group = species,
    colour = sex, facet = sex),
    rescale = TRUE,
    size = 1, interactive = FALSE,
    use.label = TRUE) +
  scale_color_manual(values = cbp1) +
  scale_fill_manual(values = cbp1) +
  theme_bw()
```

```
scale_y_discrete(breaks = NULL) + # don't show ticks  
  labs(  
    title = "Radar/spider/star chart",  
    subtitle = "Body mass of male & female penguins per species",  
    caption = "Source: https://github.com/allisonhorst/palmerpenguins")
```

## Radar/spider/star chart

Body mass of male & female penguins per species



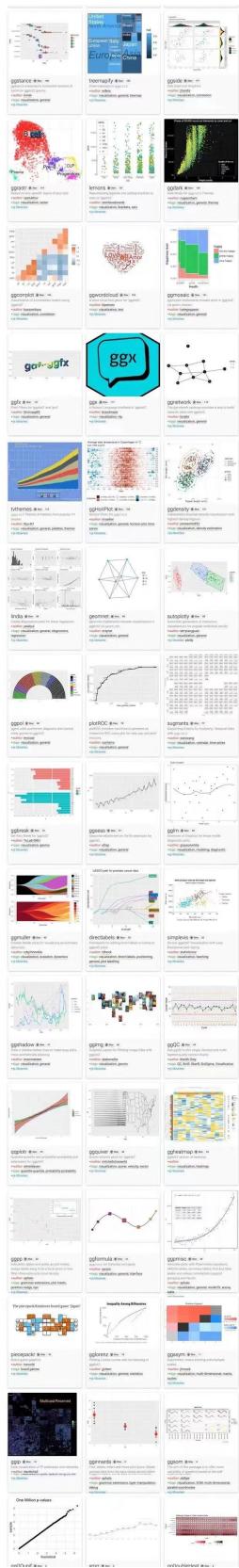
Source: <https://github.com/allisonhorst/palmerpenguins>

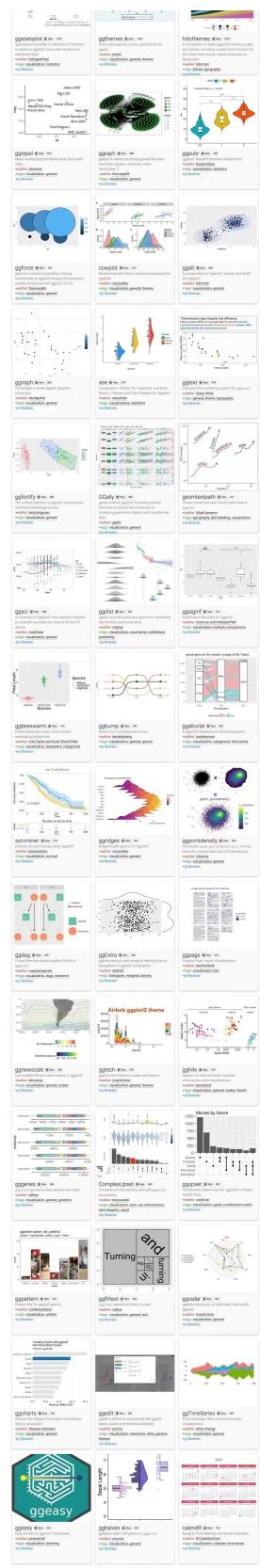
# Chapter 6

## 其他相关拓展包

### 6.1 ggplot2 官网 109 种拓展包

该网站一共汇总了 109 种与 ggplot2 包相关的拓展包，并且在不断更新中。拓展包与 ggplot2 包可以无缝连接，网站截图如下：





## 6.2 整理的相关扩展包

小编整理了些 R 语言可视化相关的拓展包。其中大部分是来自于 ggplot2 官网拓展包，也有部分不在这范围内，但是平常使用较多的包。

这里给出各包介绍的链接，主要发布于公众号中，也有的内容转载与其他公众号。由于各个包都在不断更新，可能内部函数、功能存在一些变化。如果有能力的读者，建议去查阅最新的官方消息。

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 ggpublisher 包制图
10. reticulate 包—数据科学者的福音
11. igraph 包—绘制网络图
12. R-ggchicklet - 圆角条形图绘制
13. ggstatsplot 包：数据可视化与统计建模相结合
14. igraph 包——绘制网络图
15. explore 包——化繁为简，交互式数据探索分析
16. grafify 包 | 快速绘制 19 个 ggplot2 图形
17. ggThemeAssist 包：交互式修改图形主题

# Chapter 7

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

## 7.1 leaflet 包

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

## 7.2 dygraphs 包

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

## 7.3 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)

```

## 7.4 DT 包

```

library(DT)
datatable(iris)

```

## 7.5 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)

```

## 7.6 利用 Shiny 包实现可交互的 Web 应用

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 8

## 常用案例

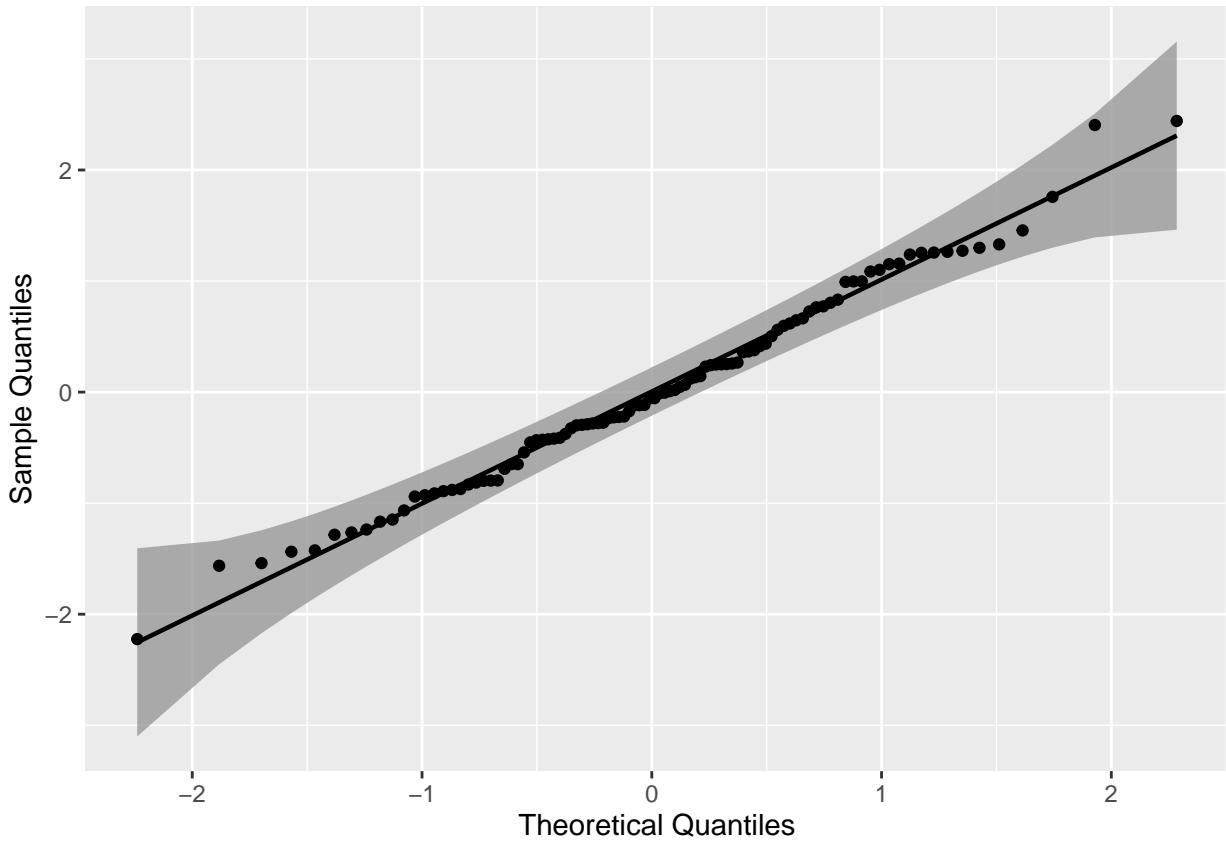
### 8.1 绘制不同分布的 QQ 图

这里主要是用 qqplotr 包进行绘制，参考的博客：An Introduction to qqplotr。

#### 8.1.1 简单版本

##### 8.1.1.1 绘制正态分布的 QQ 图

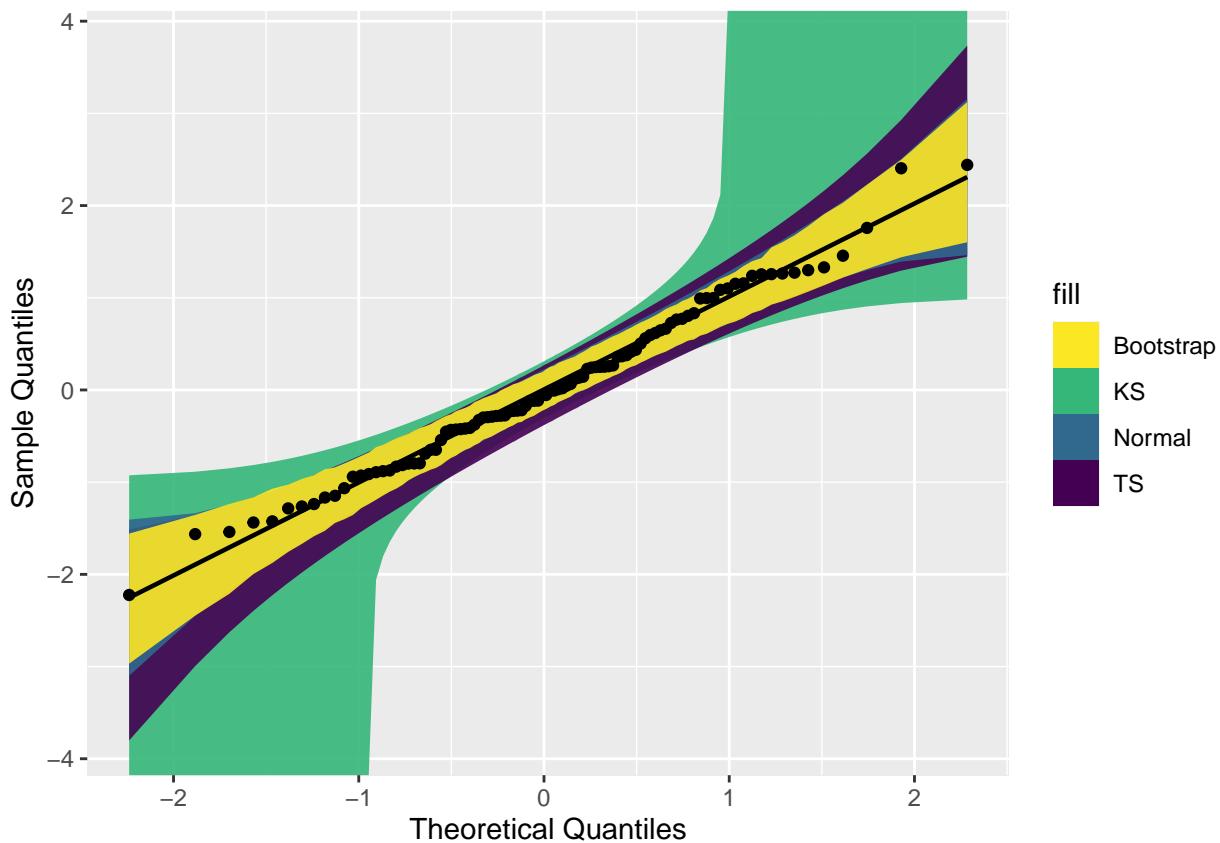
```
library(qqplotr)
library(ggplot2)
# 随机产生数据
set.seed(0)
smp <- data.frame(norm = rnorm(100))
# 绘制
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
  stat_qq_band() +
  stat_qq_line() +
  stat_qq_point() +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg
```



使用三种不同的置信带构造置信区间，其可以用参数 `bandType` 调整。

```
library(viridis)
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
  geom_qq_band.bandType = "ks", mapping = aes(fill = "KS"), alpha = 0.9) +
  geom_qq_band.bandType = "ts", mapping = aes(fill = "TS"), alpha = 0.9) +
  geom_qq_band.bandType = "pointwise", mapping = aes(fill = "Normal"), alpha = 0.9) +
  geom_qq_band.bandType = "boot", mapping = aes(fill = "Bootstrap"), alpha = 0.9) +
  stat_qq_line() +
  stat_qq_point() +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
  scale_fill_viridis(discrete = T, direction = -1)

gg
```



### 8.1.2 进阶版本

读者绘制正态分布的 QQ 图，还是比较简单的。但是如果其他分布的情况呢？

这里以一个可靠性数据为例子，该数据来源于文献：Badar, M. G., Priest, A. M. (1982). Statistical aspects of fiber and bundle strength in hybrid composites. In: Hayashi, T., Kawata, K., Umekawa, S., eds. Progress in Science and Engineering Composites. Tokyo: ICCM-IV, pp. 1129–1136。

```
data = data.frame('y' = c(1.339, 1.434, 1.549, 1.574, 1.589, 1.613, 1.746, 1.753, 1.764, 1.807, 1.812))
```

#### 8.1.2.1 绘制指数分布的 QQ 图

这里我们绘制其指数分布的 QQ 图。根据指数函数参数拟合该数据之后，得到 `rate = 2.2867`，并将其保存到 list 中。

具体如何拟合，读者自行搜索 R 包中的相关函数。

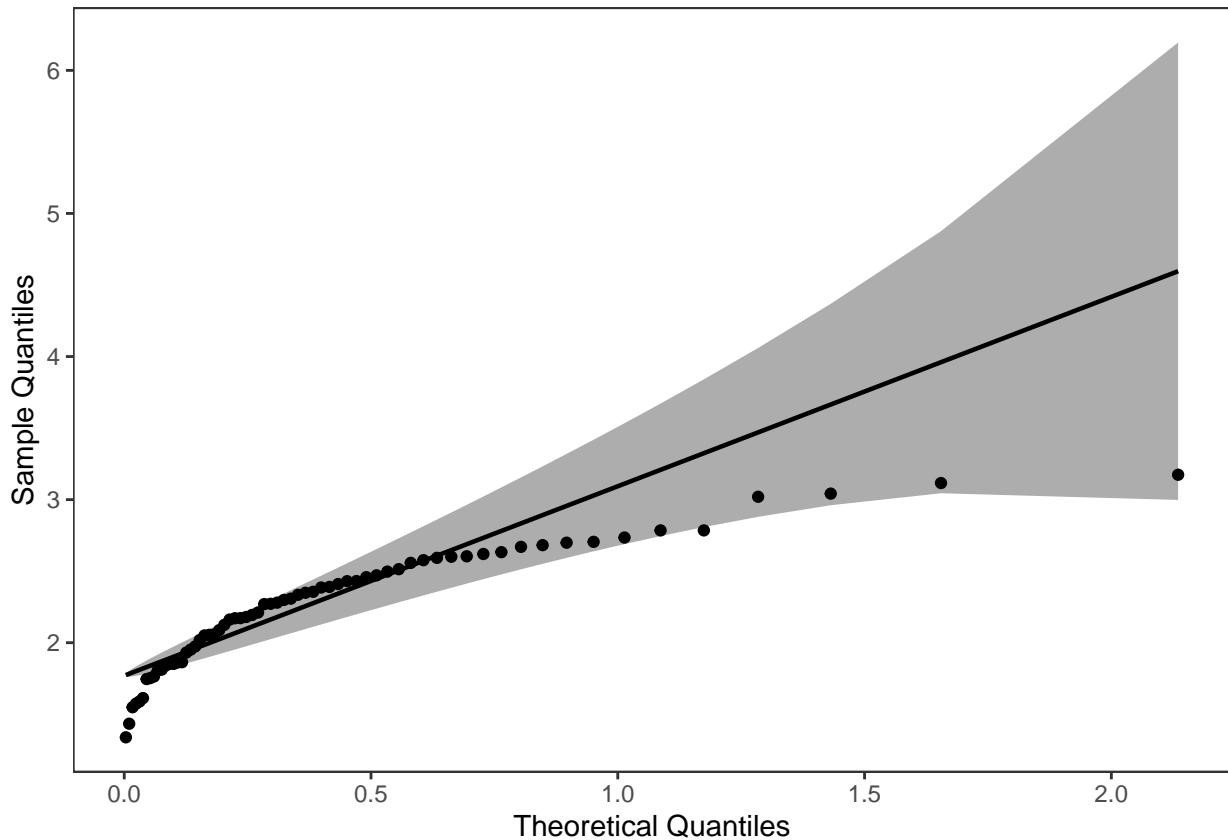
其他代码基本不变，主要是将 `stat_qq_line()` 和 `stat_qq_point()` 中的分布设定下，参数设定下。

```
# exponential distribution
dp <- list(rate = 2.2867)
```

```

di <- "exp"
p1 = ggplot(data = data, mapping = aes(sample = y)) +
  stat_qq_band(distribution = di, dparams = dp) +
  stat_qq_line(distribution = di, dparams = dp) +
  stat_qq_point(distribution = di, dparams = dp) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
  theme_bw() +
  theme(panel.grid = element_blank())
p1

```



### 8.1.2.2 绘制威布尔分布的 QQ 图

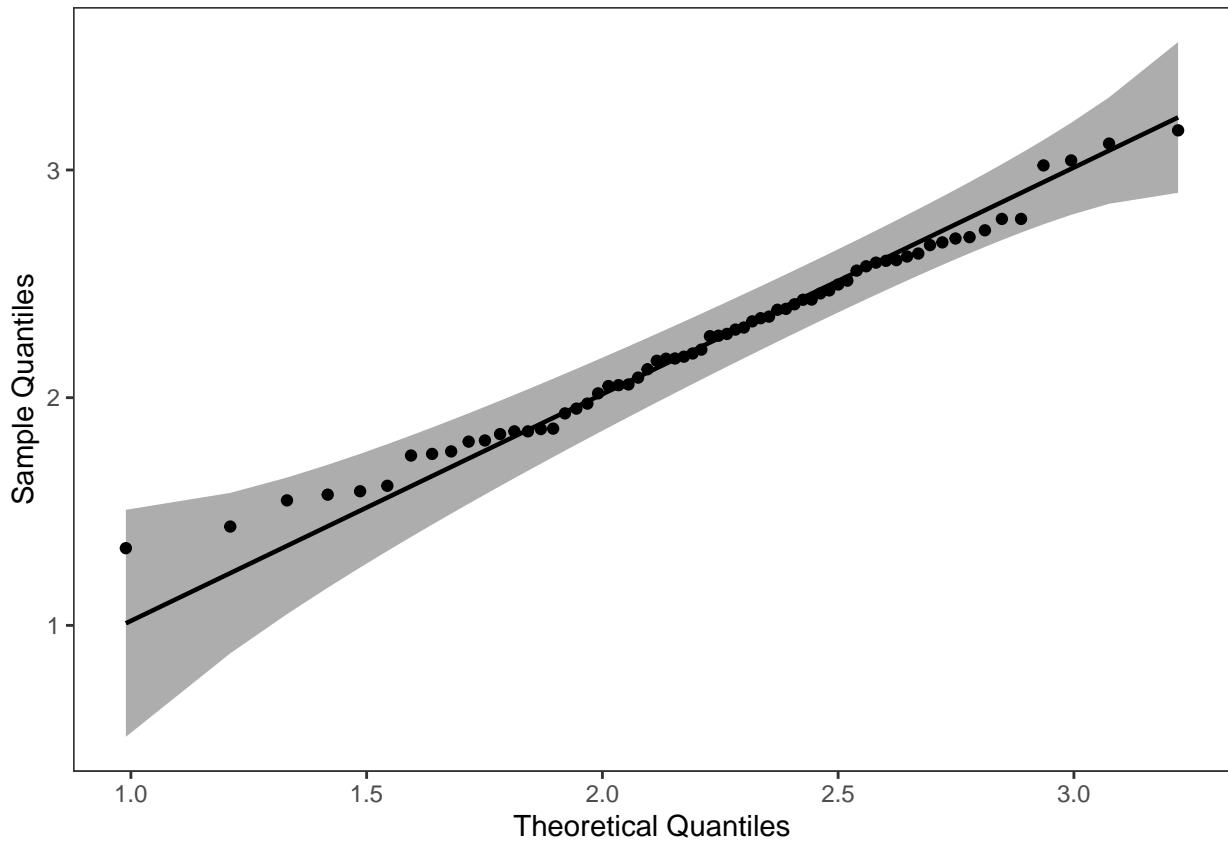
同理，将该数据应用到威布尔分布中。结果如下：

```

# weibull distribution
di <- "weibull" # exponential distribution
dp <- list(shape=5.4766,scale=2.4113)
p2 = ggplot(data = data, mapping = aes(sample = y)) +
  stat_qq_band(distribution = di, dparams = dp) +

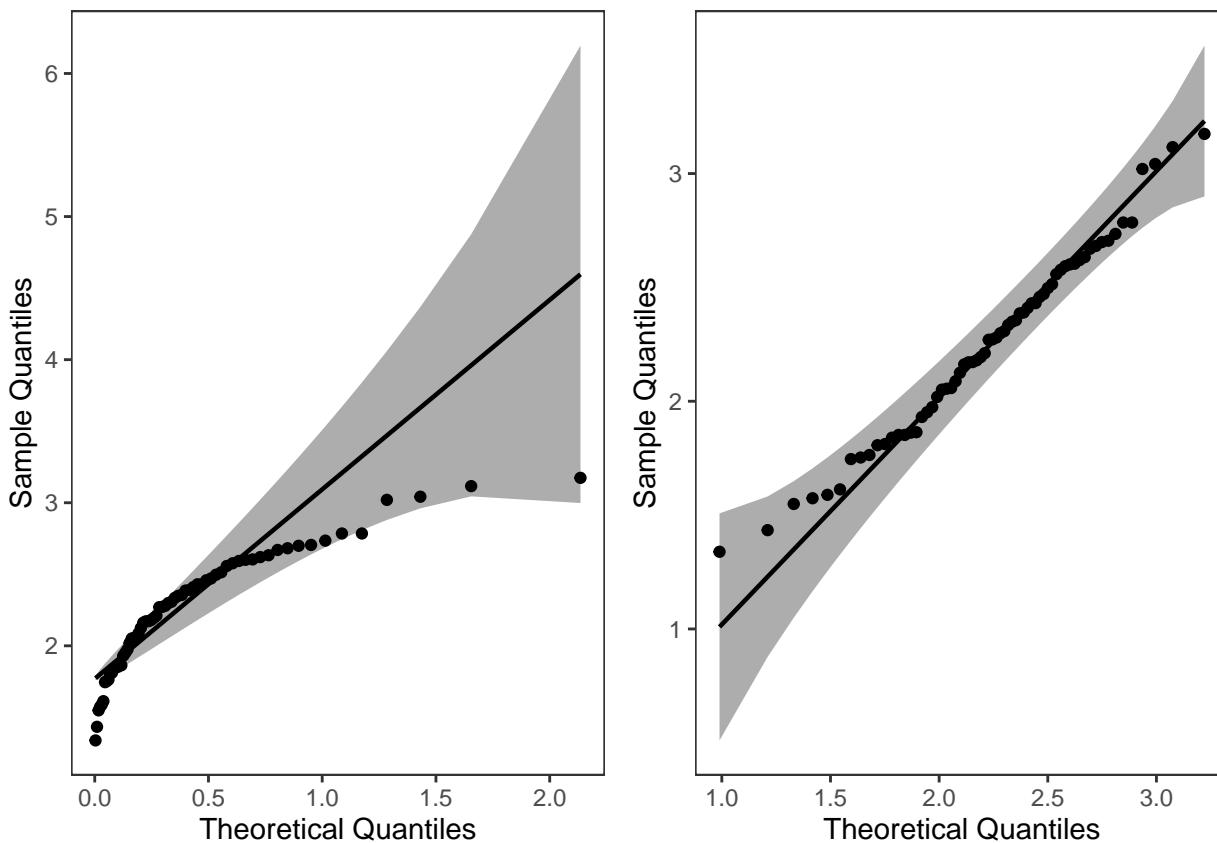
```

```
stat_qq_line(distribution = di, dparams = dp) +  
stat_qq_point(distribution = di, dparams = dp) +  
labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +  
theme_bw() +  
theme(panel.grid = element_blank())  
p2
```



可以看到该数据集，更适合使用 weibull 分布进行拟合。

```
library(cowplot)  
plot_grid(p1, p2, ncol = 2, nrow = 1)
```



## 8.2 绘制混合密度函数图以及添加分位数线

主要使用 `ggridges` 包中的 `stat_density_ridges()`。参考的博客：<https://cran.r-project.org/web/packages/ggridges/vignettes/introduction.html>

### 8.2.1 加载包

```
library(ggplot2)
library(ggridges)
```

### 8.2.2 产生数据集

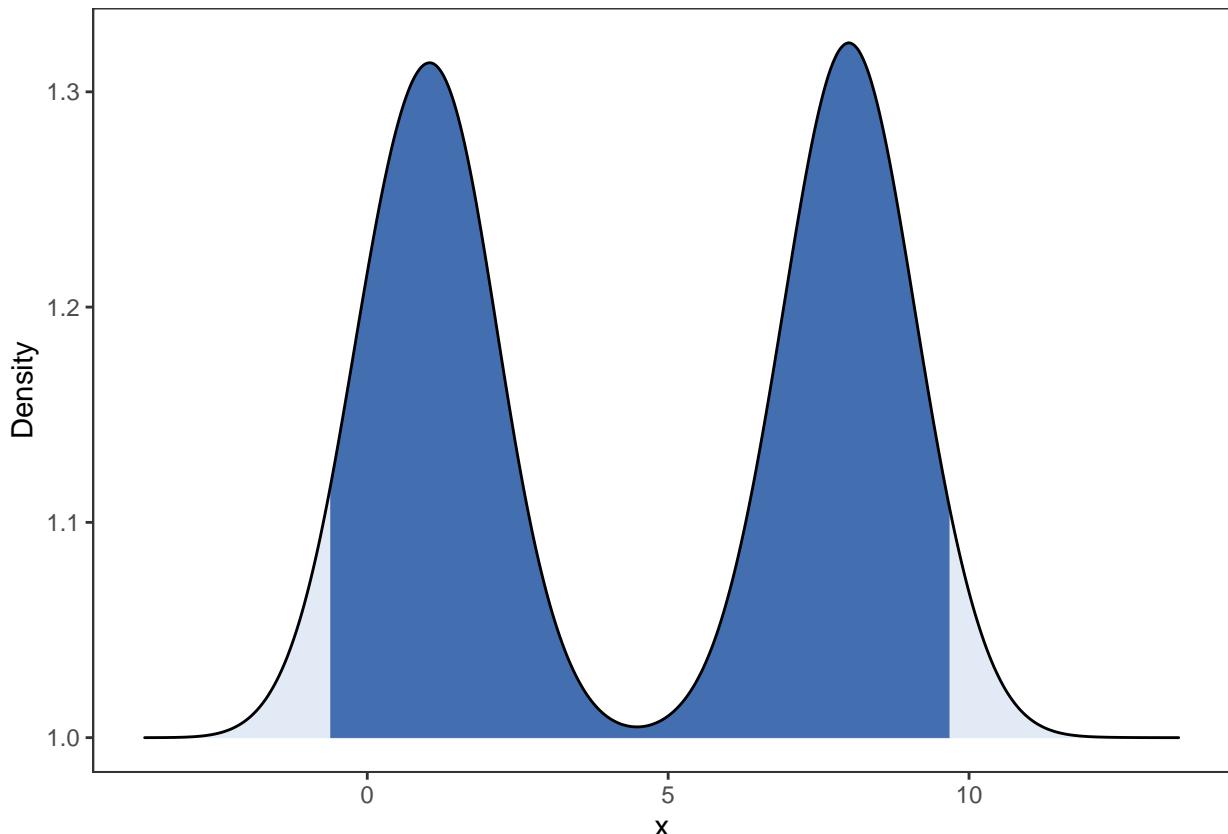
假设数据来源于一个混合分布。

```
item <- 10000
inds <- rbinom(1, item, 0.5)
x <- c(rnorm(inds, 1, 1), rnorm(item - inds, 8, 1))
data <- data.frame("value" = x, "class" = rep(1, length(x)))
```

### 8.2.3 绘制密度函数图并添加分位数线

```
# 绘图
```

```
p1 <- ggplot(data, aes(x = value, y = class, fill = factor(stat(quantile)))) +  
  stat_density_ridges(  
    geom = "density_ridges_gradient",  
    calc_ecdf = TRUE,  
    quantiles = c(0.025, 0.975)  
  ) +  
  scale_fill_manual(  
    name = "Probability", values = c("#E2EAF6", "#436FB0", "#E2EAF6")  
  ) +  
  theme_bw() +  
  theme(legend.position = "none", panel.grid = element_blank()) +  
  labs(x = "x", y = "Density")  
p1
```



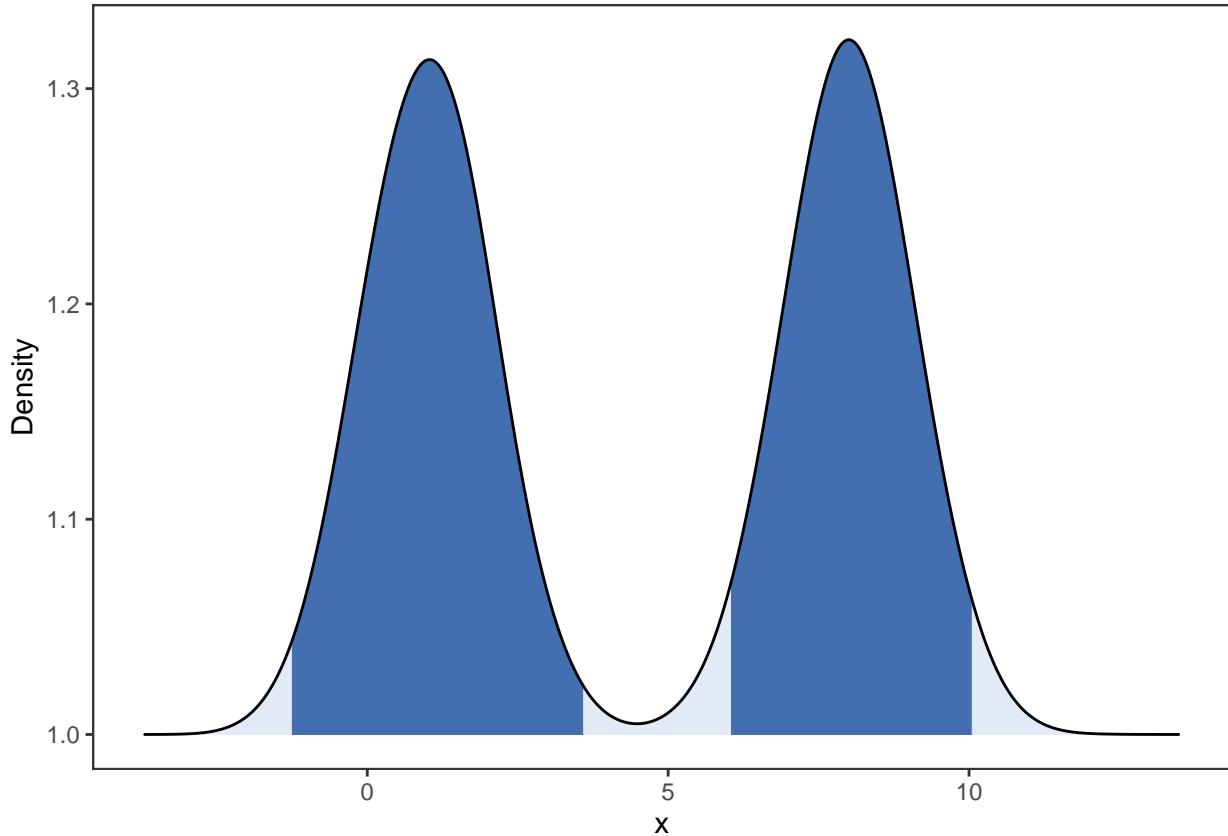
```
p2 <- ggplot(data, aes(x = value, y = class, fill = factor(stat(quantile)))) +  
  stat_density_ridges(
```

```

geom = "density_ridges_gradient",
calc_ecdf = TRUE,
quantiles = c(0.005, 0.495, 0.51, 0.99)
) +
scale_fill_manual(
  name = "Probability", values = c("#E2EAF6", "#436FB0", "#E2EAF6", "#436FB0", "#E2EAF6"),
) +
theme_bw() +
theme(legend.position = "none", panel.grid = element_blank()) +
labs(x = "x", y = "Density")

```

p2

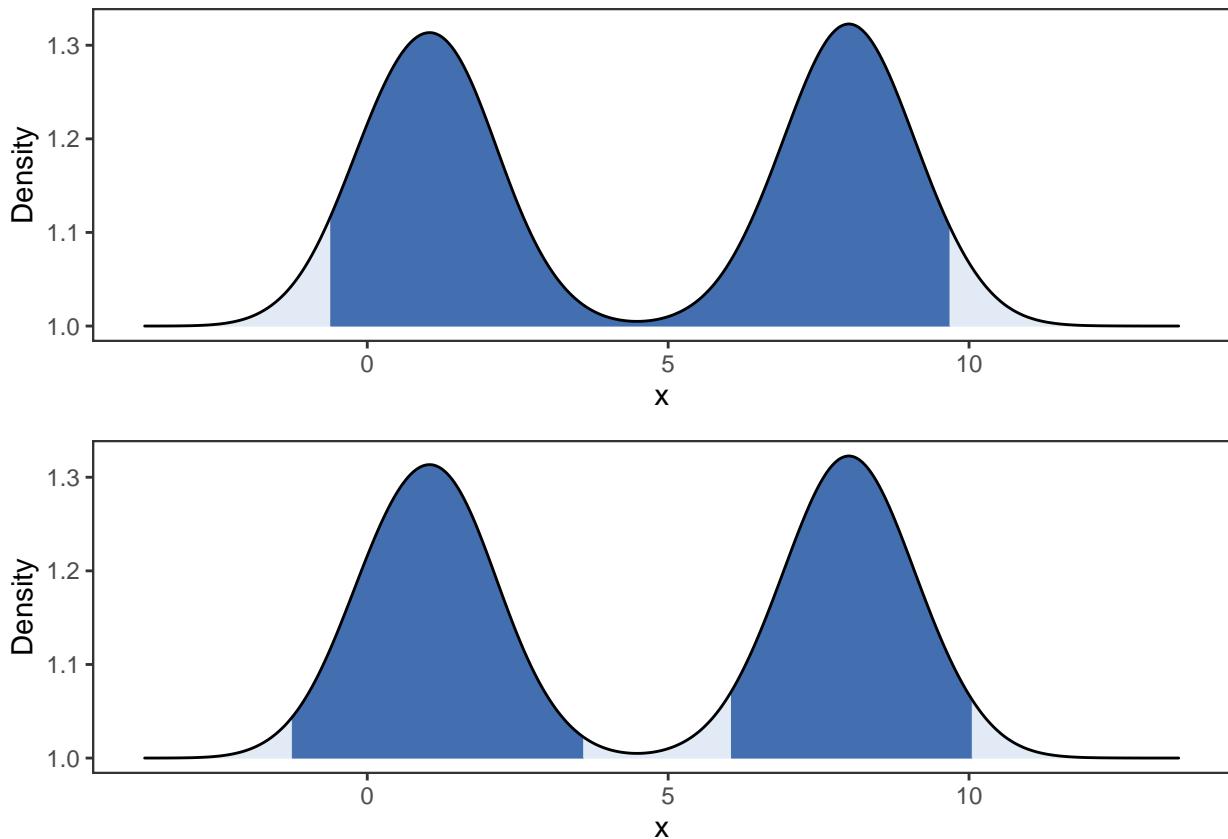


#### 8.2.4 合并两图（使用 cowplot 包）

```

library(cowplot)
# pdf("plot_cow.pdf", width = 8, height = 4)
plot_grid(p1, p2, ncol = 1, nrow = 2)

```



```
# dev.off()
```

## 8.3 绘制不同参数的密度函数

### 8.3.1 Beta 密度函数

绘制不同参数下的 Beta 密度函数。具体该分布的介绍可见：[https://en.wikipedia.org/wiki/Beta\\_distribution](https://en.wikipedia.org/wiki/Beta_distribution)

Beta 函数如下：

$$B(a, b) = \int_0^1 x^{a-1} (1-x)^{b-1} dx, \quad a, b > 0$$

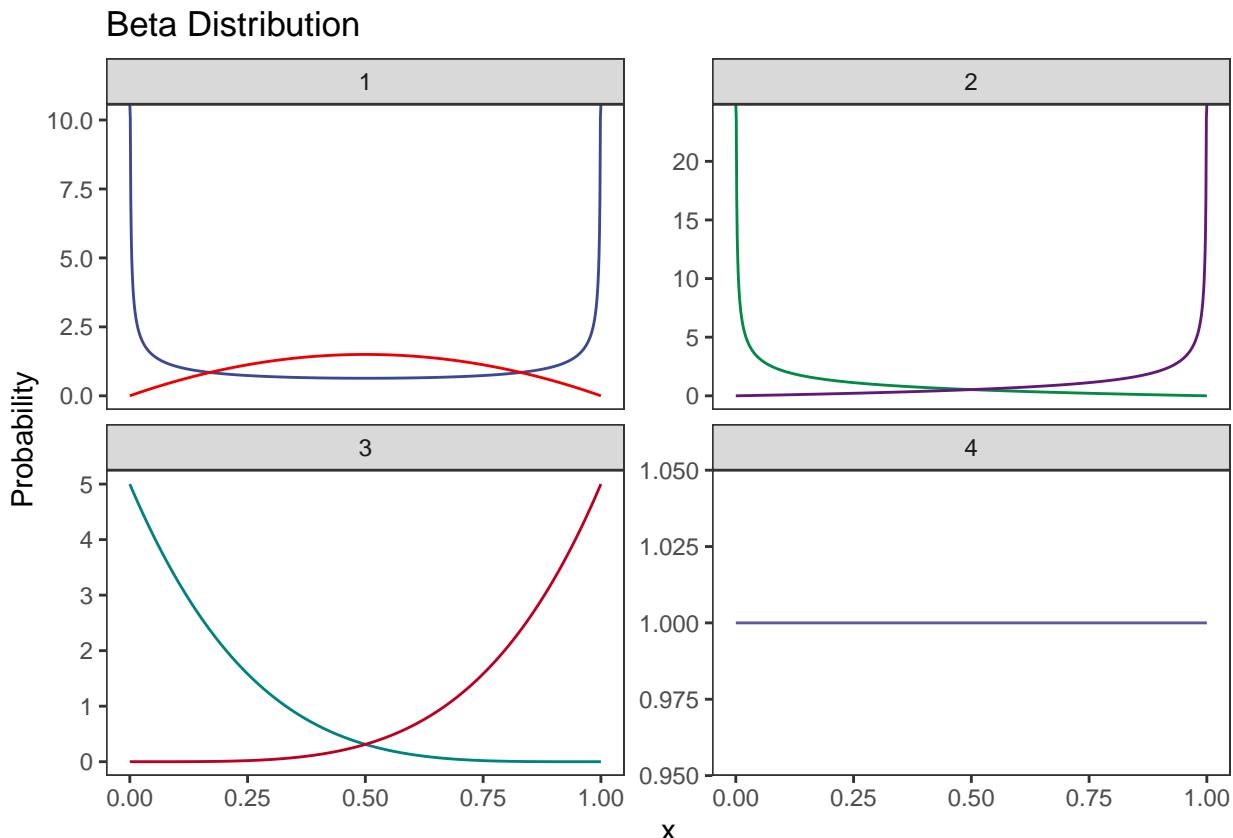
Beta 分布：

随机变量 X 的密度函数为：

$$p(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

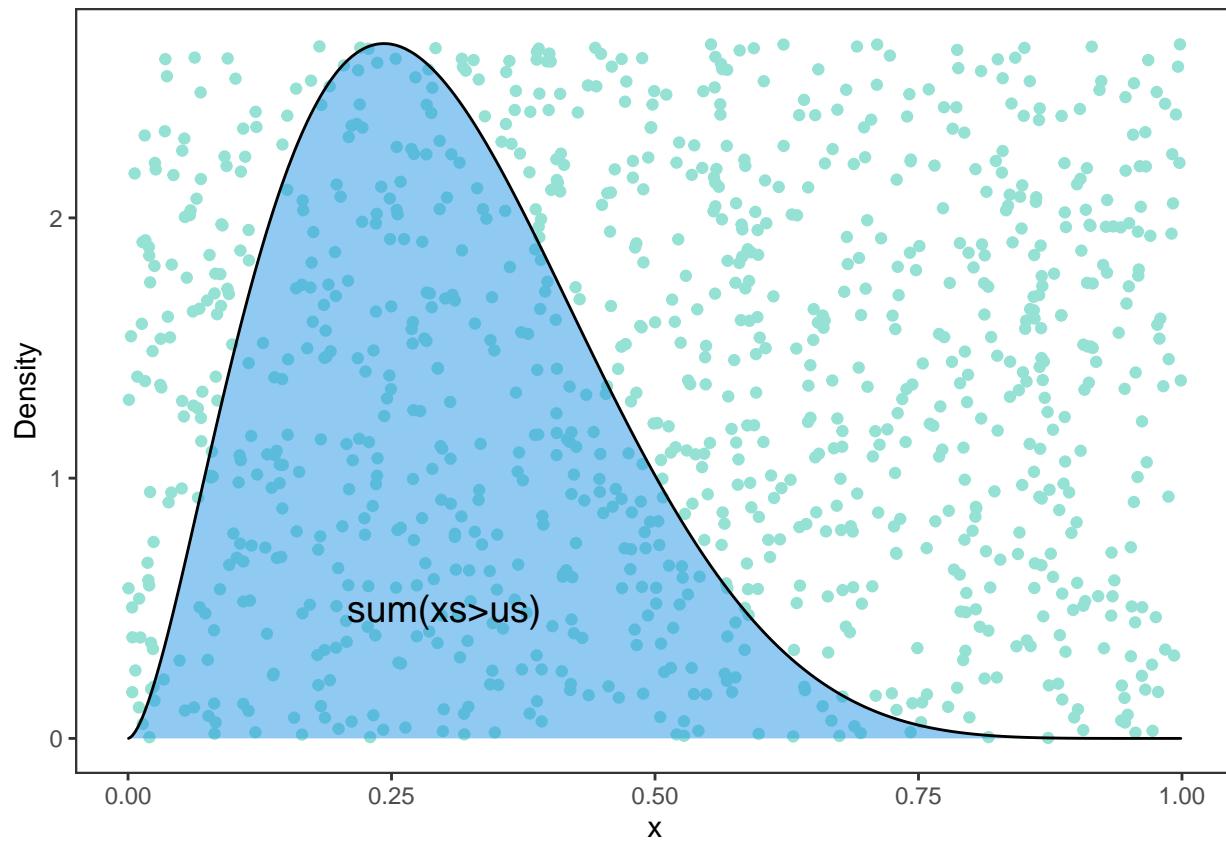
存在问题：还未把各个密度函数的参数加入。

```
## Beta Distribution
library(ggplot2)
library(reshape2)
library(ggsci)
item = 1000
x <- seq(0,1,length = item)
beta_dist <- data.frame(cbind(x,dbeta(x,0.5,0.5), dbeta(x,2,2), dbeta(x,0.5,2),dbeta(x,2,0.5),dbeta(x,1,1)))
colnames(beta_dist) <- c("x","a=0.5,b=0.5","a=2,b=2","a=0.5,b=2",
                         "a=2,b=0.5","a=1,b=5","a=5,b=1","a=1,b=1")
beta_dist <- melt(beta_dist,x)
beta_dist$class" = c(rep(1:3,each = 2*item),rep(4,item))
g <- ggplot(beta_dist, aes(x,value, color=variable))
g + geom_line() +
  facet_wrap(vars(class),scales = "free_y") +
  labs(title="Beta Distribution", x="x", y="Probability") +
  scale_color_aaas() +
  theme_bw() + theme(panel.grid = element_blank(),legend.position = 'none')
```



## 8.4 散点图中加入第三变量的密度函数

```
### Accept-reject algorithm
### beta(2.7,6.3)
library(ggplot2)
num <- 1000
ys <- runif(num)
us <- runif(num, 0, 2.67)
xs <- dbeta(ys, 2.7, 6.3)
data <- data.frame("x_rand" = ys, "Density" = us, "true_x" = xs)
ggplot(data) +
  geom_point(aes(x_rand, Density), col = "#95E1D3") +
  geom_area(aes(x_rand, xs),
            fill = 4, alpha = 0.5,
            color = 1, # Line color
            lwd = 0.5, # Line width
            linetype = 1
  ) +
  annotate("text", x = 0.3, y = 0.5, label = "sum(xs>us)", size = 5) +
  labs(x = "x") +
  theme_bw() +
  theme(panel.grid = element_blank())
```



## 8.5 绘制 Gamma 分布相关图形

- 自定义主题

```
library(ggridges)
# 自定义主题
theme_manual = function(){
  theme(panel.grid = element_blank(),
        panel.border = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        legend.position="none")
}
```

- 数据模拟

```
library(DT)
library(ggplot2)
num = 100
```

```
data = data.frame('id' = 1:num, 'value' = rgamma(num, 2, 1), 'class' = factor(gl(5, num/5)))
datatable(data)
```

Show  entries Search:

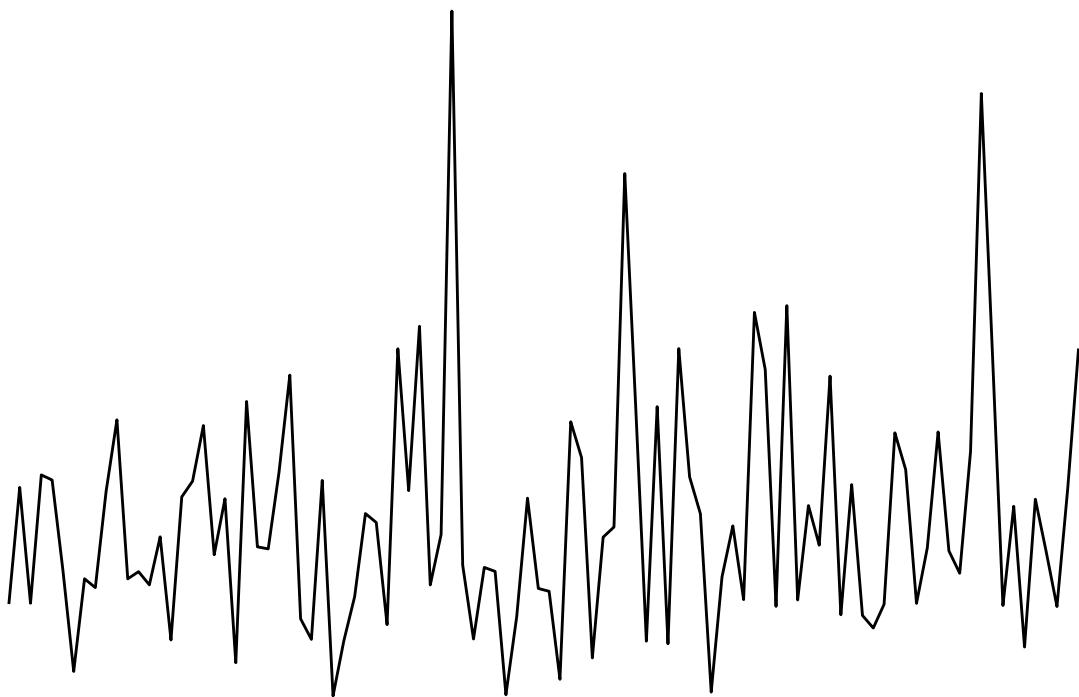
	id	value	class
1	1	1.08910568471399	1
2	2	2.3924841180044	1
3	3	1.09812011053858	1
4	4	2.5324710486822	1
5	5	2.47536196011498	1
6	6	1.46782830106023	1
7	7	0.33584492113963	1
8	8	1.37133001028442	1
9	9	1.27270852651137	1
10	10	2.35078221168434	1

Showing 1 to 10 of 100 entries Previous      ...  Next

- 绘图

```
p1 = ggplot(data,aes(id,value)) +
  geom_line() +
  geom_hline(yintercept = max(data$value)+0.2) +
  geom_hline(yintercept = min(data$value)-0.2) +
  theme_bw() +
  theme_manual() + xlab('') + ylab('')
```

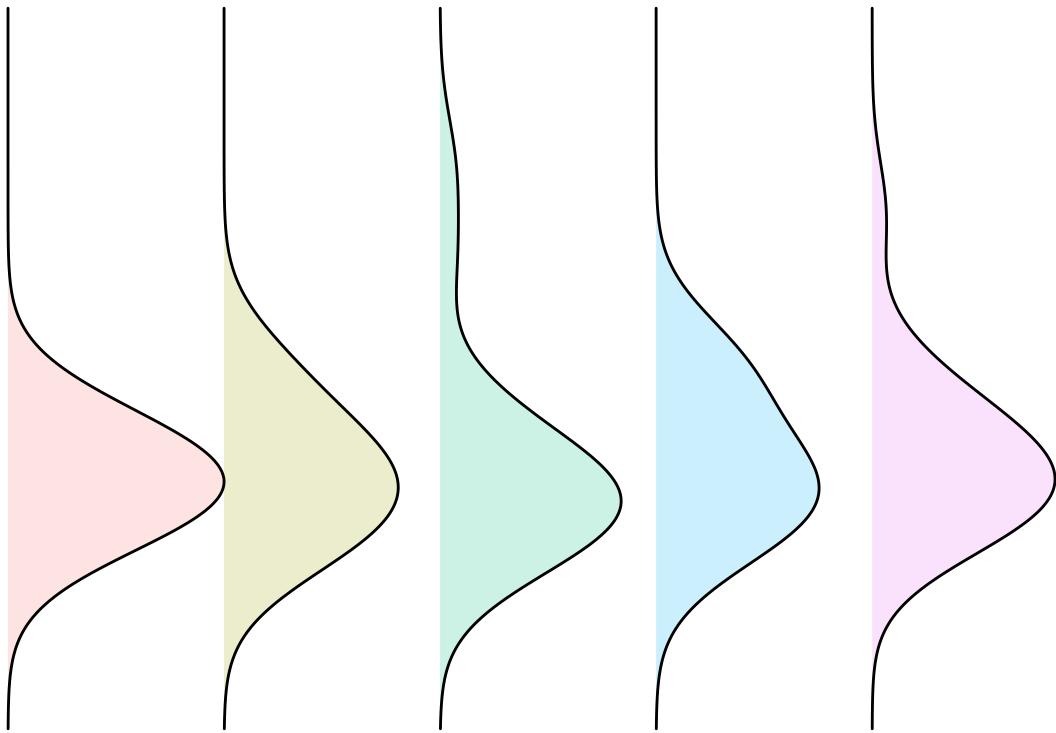
p1



```
## 纵向密度函数图
```

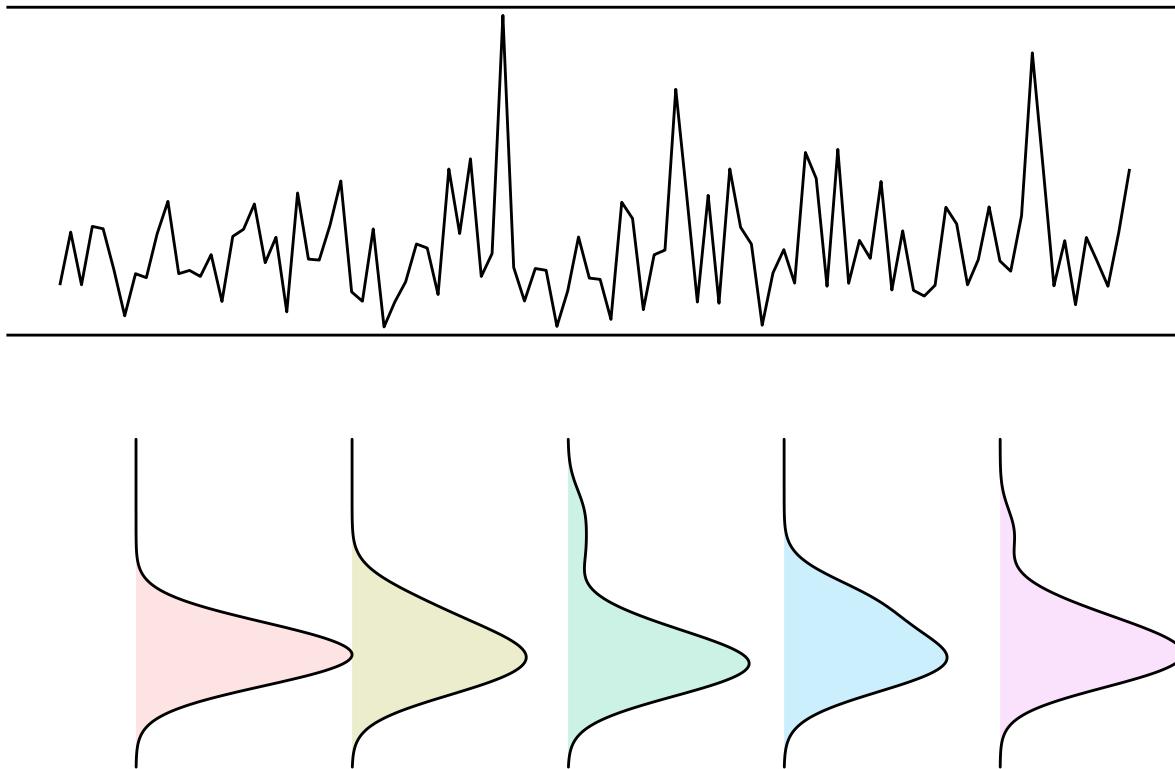
```
p2 = ggplot(data, aes(x = value, y = class, fill = class)) +
  geom_density_ridges(aes(), alpha = 0.2, scale = 1, bandwidth = 1) +
  coord_flip() +
  # scale_point_color_hue(l = 40) +
  # scale_discrete_manual(aesthetics = "point_shape", values = c(21, 22, 23, 24, 25)) +
  theme_bw() +
  theme_manual() + xlab('') + ylab('')
```

p2



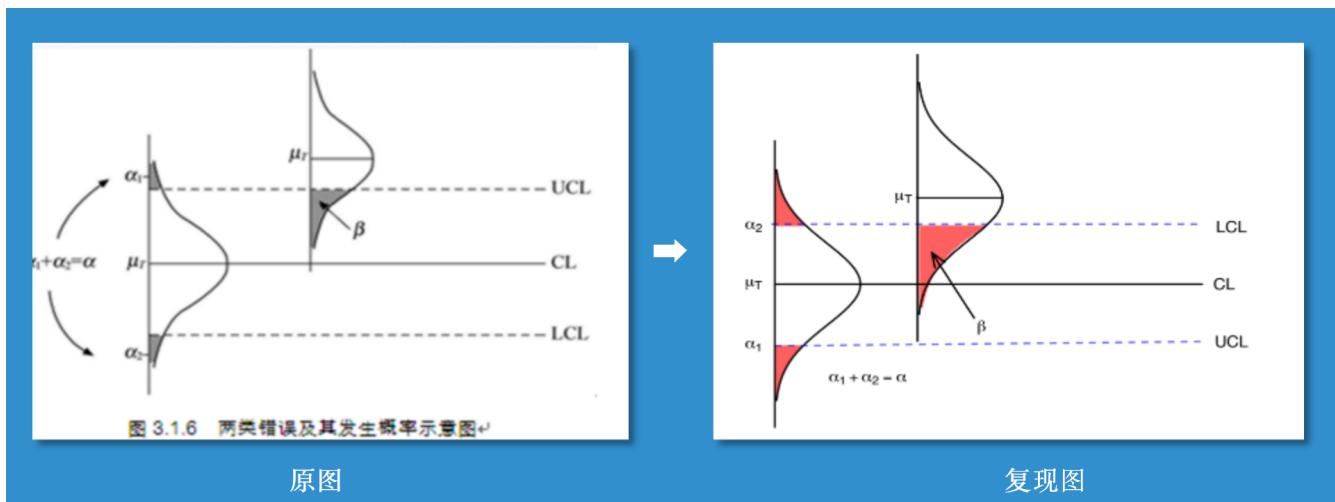
- 合并图形

```
library(cowplot)  
plot_grid(p1,p2,ncol = 1)
```



## 8.6 纯手动复现密度图

老板觉得课件的图形太过模糊和单调，于是想让我用 ggplot2 包复现一下结果，做的更加高清、精美些。于是花将近一个小时复现了下面这个图形：



主要有两部分构成：1. 绘制密度函数曲线并填充分位数面积；2. 添加各种线段和文字。

- 数据产生

```
# 数据产生
set.seed(1)
mu = c(2,5)
std = c(1,1)
num = 1000
r1 = rnorm(num,mu[1],std[1])
r2 = rnorm(num,mu[2],std[2])
data = data.frame('value' = c(r1,rep(NA,num),r2,rep(NA,5*num)),
                  'class' = factor(rep(c(1:8),each= num)))
knitr::kable(head(data))
```

value	class
1.373546	1
2.183643	1
1.164371	1
3.595281	1
2.329508	1
1.179532	1

- 画图

文中还是使用前面自定义的主题。

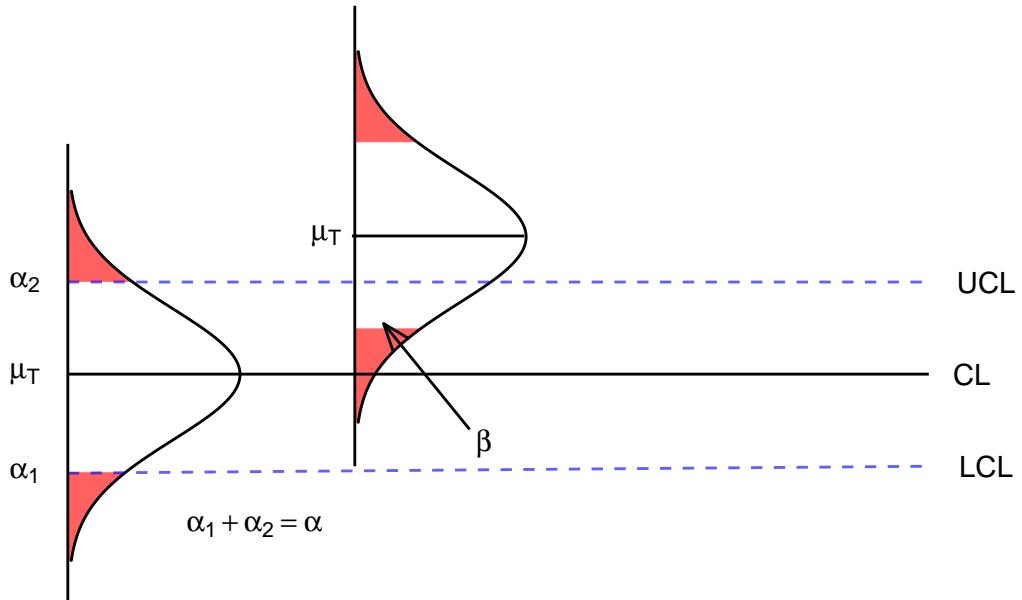
```
# 画图
```

```
ggplot(data, aes(x = value, y = class,fill = factor(stat(quantile)))) +
# 添加密度函数图
stat_density_ridges(
  geom = "density_ridges_gradient",
  calc_ecdf = TRUE,rel_min_height = 0.02,
  quantiles = c(0.025, 0.975),
  alpha = 1,scale = 0.6,bandwidth = 1
) +
scale_fill_manual(
  name = "Probability", values = c("#FF0000A0", "white", "#FF0000A0"), "##0000FFA0"
  labels = c("(0, 0.025]", "(0.025, 0.975]", "(0.975, 1]")
) +
# 添加各种线段, 文字
annotate("segment", x = mu[1], xend = mu[1], y = 1, yend = 7,colour = "black") +
annotate("segment", x = mu[2]-1, xend = mu[2]-1, y = 1, yend = 7,colour = "#0000FFA0",lty = "dashed")
annotate("segment", x = mu[1]-2.15, xend = mu[1]-2, y = 1, yend = 7,colour = "#0000FFA0",lty = "dashed")
```

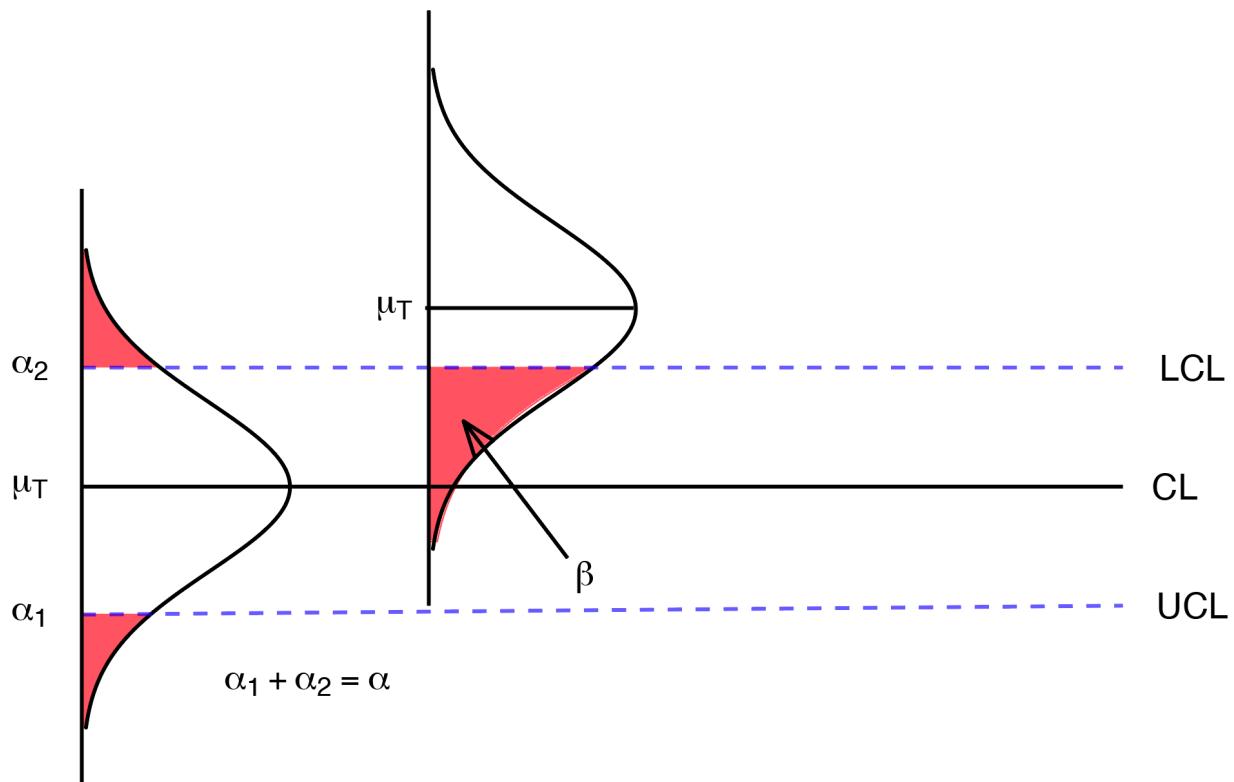
```

annotate("segment", x = mu[2], xend = mu[2], y = 3-0.02, yend = 4.18, colour = "black") +
annotate("segment", x = 0, xend = 10, y = 3, yend = 3, colour = "black") +
annotate("segment", x = -3, xend = 7, y = 1, yend = 1, colour = "black") +
annotate("text", x = mu[1], y = 0.7, label = expression(mu[T])) +
annotate("text", x = mu[1]-2.15, y = 0.7, label = expression(alpha[1])) +
annotate("text", x = mu[1]+2, y = 0.7, label = expression(alpha[2])) +
annotate("text", x = mu[2], y = 2.8, label = expression(mu[T])) +
annotate("text", x = 0.5, y = 3.9, label = expression(beta)) +
annotate(geom = "line", x = c(0.8, 3.1),
         y = c(3.8, 3.2),
         arrow = arrow(angle = 20, length = unit(4, "mm")))) +
annotate("text", x = mu[1], y = 7.3, label = "CL") +
annotate("text", x = mu[2]-1, y = 7.4, label = "UCL") +
annotate("text", x = mu[1]-2, y = 7.4, label = "LCL") +
annotate("text", x = -1.3, y = 2.3, label = expression(alpha[1] + alpha[2] == alpha)) +
coord_flip() +
theme_bw() +
theme_manual() + xlab('') + ylab('')

```



可以看到，现在得到的结果还是有点问题的。笔者能力有限，不能使用代卖复现的一模一样（也是为了省时间），于是我是用了 AI 大法，对该图形进行了修改。最后得到：



## 8.7 绘制贝叶斯分析相关图形

主要使用 bayesplot 包进行绘制，对应的速查表网址为：<https://raw.githubusercontent.com/rstudio/cheatsheets/main/bayesplot.pdf>

```
# 加载需要使用的包
library(showtext)
showtext_auto() # 解决中文字体乱码问题
library(bayesplot)
library(rstanarm)
options(mc.cores = parallel::detectCores())
library(ggplot2)
library(dplyr)
```

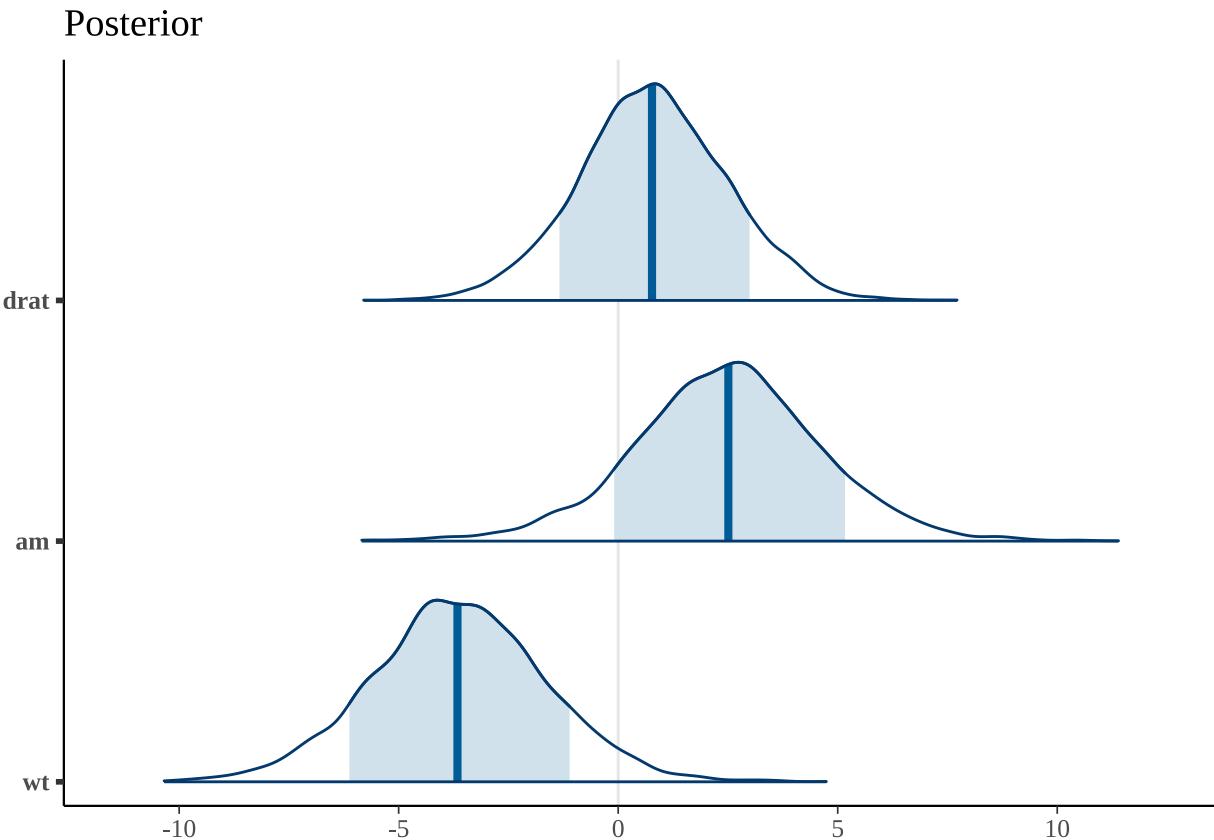
### 8.7.1 模型建模

为了展示贝叶斯图，我们将使用 `rstanarm::stan_glm` 拟合线性回归，并始终使用这个模型。

```
model <- stan_glm(mpg~., data=mtcars, chains=4)
posterior <- as.matrix(model)
# head(posterior)
```

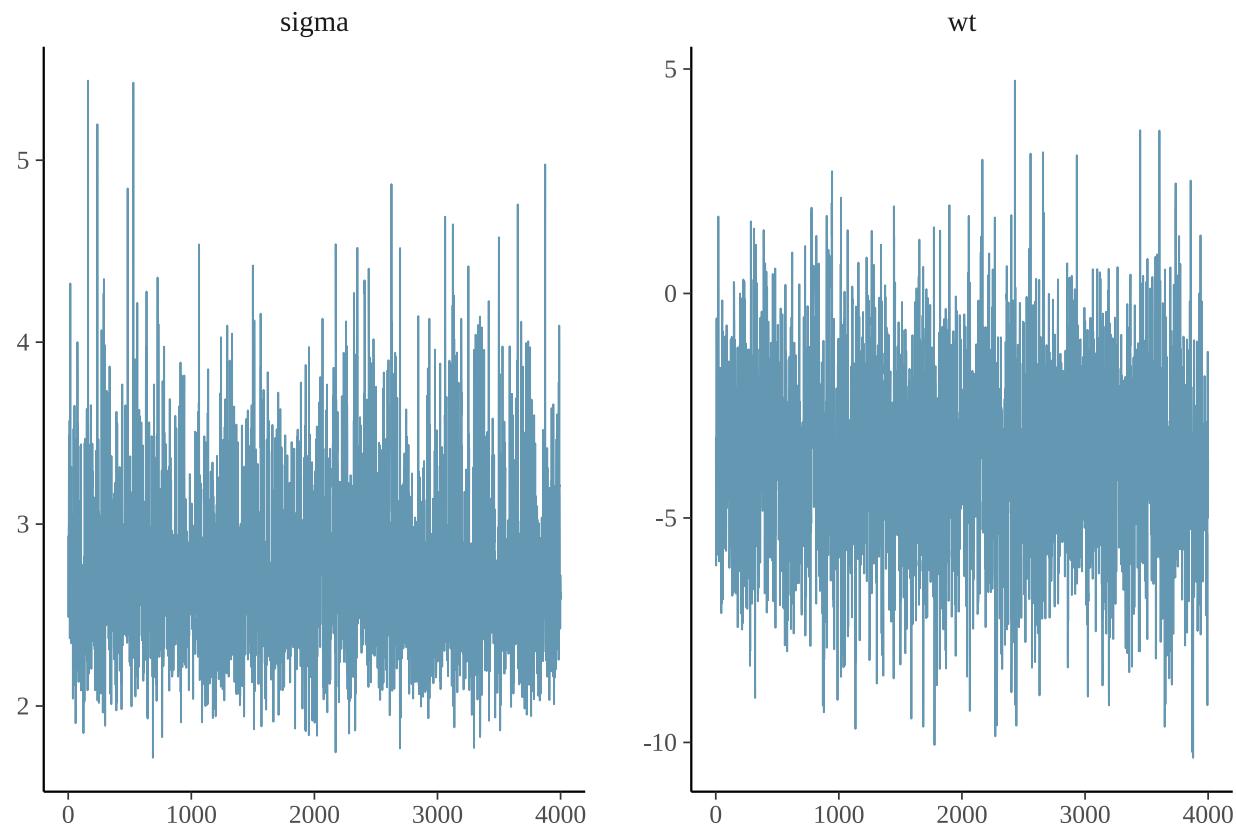
展示你感兴趣参数的后验分布。

```
showtext_auto()
mcmc_areas(posterior, pars = c("drat", "am", "wt"), prob = 0.8) + ggtitle("Posterior")
```



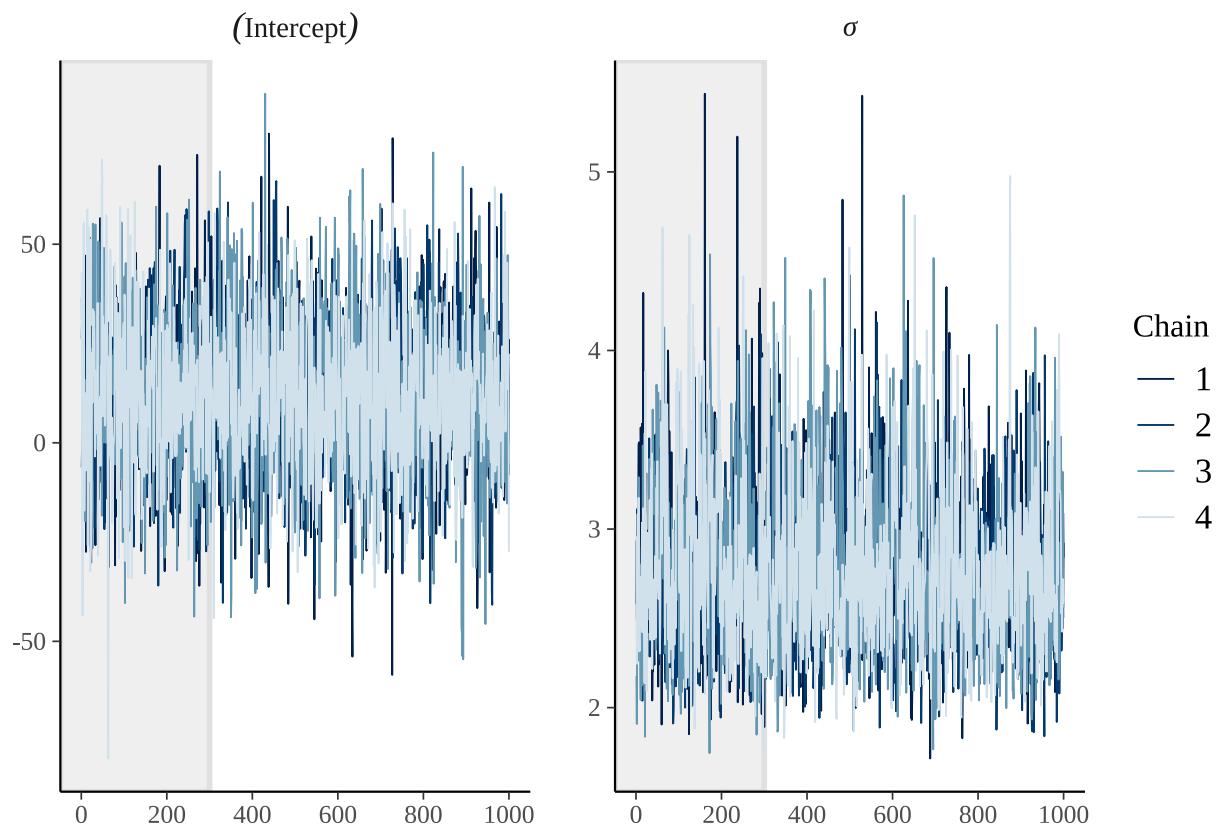
使用轨迹追踪图诊断收敛性

```
mcmc_trace(posterior, pars=c("sigma", "wt"))
```



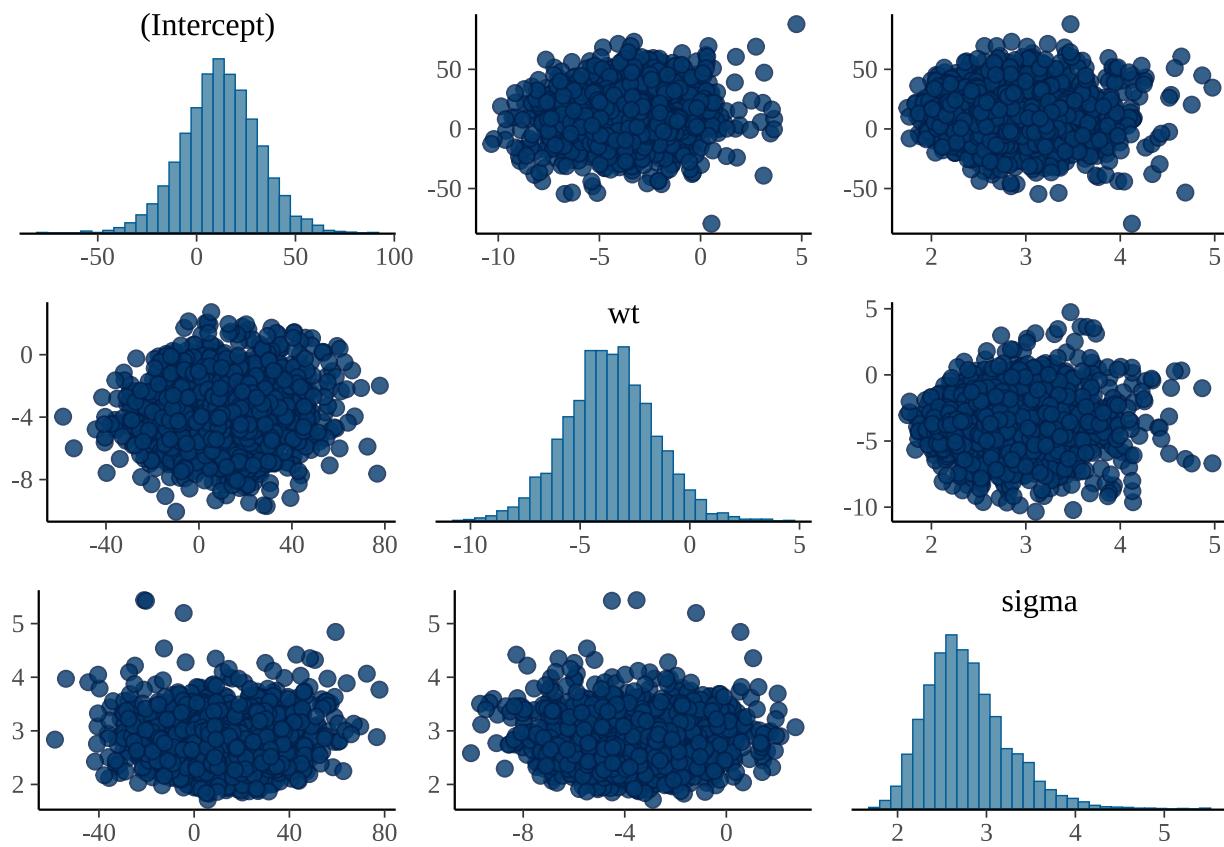
使用 `as.array`, 可以提取每一个链的数据。这允许您查看每个链所选参数的跟踪图。

```
posterior_chains <- as.array(model)
fargs <- list(ncol = 2, labeller = label_parsed)
pars <- c("(Intercept)", "sigma")
chains_trace <- mcmc_trace(posterior_chains, pars = pars, n_warmup = 300, facet_args = fargs)
chains_trace
```



配对图有助于确定是否有任何高度相关的参数。

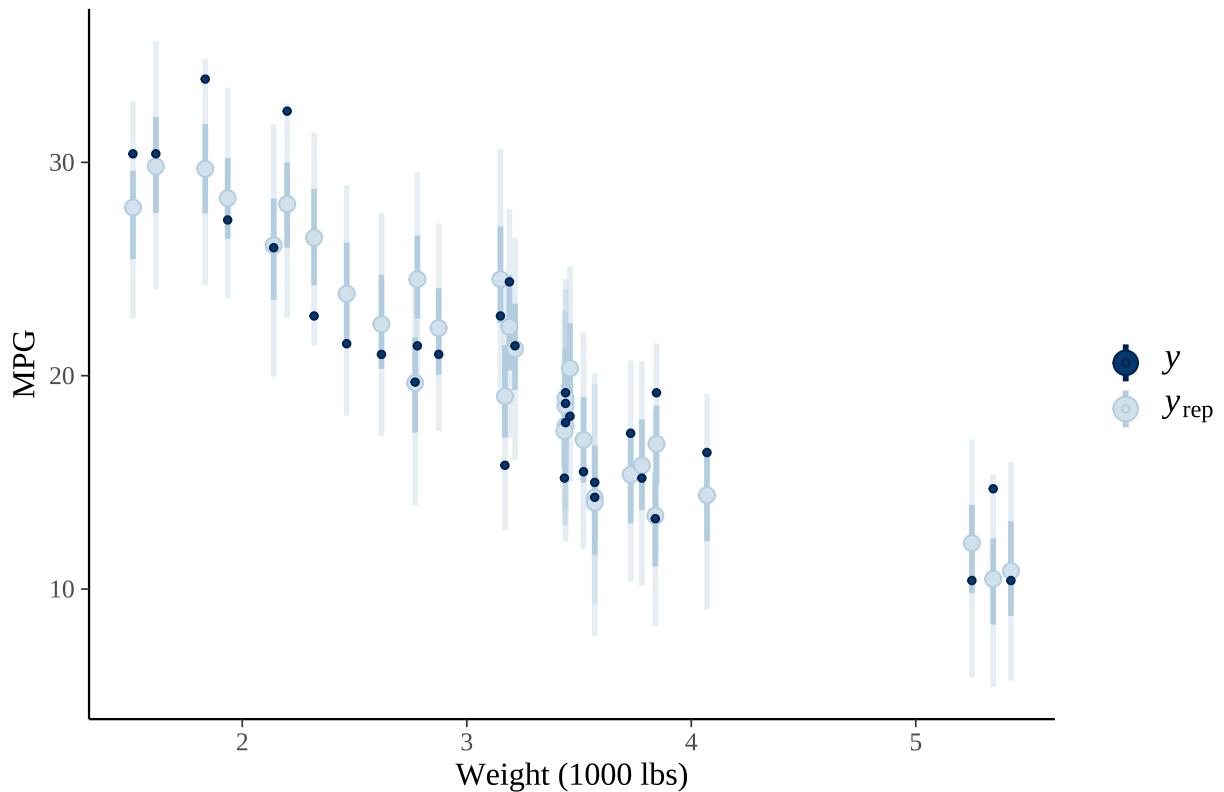
```
posterior_chains %>% mcmc_pairs(pars = c("(Intercept)", "wt", "sigma"))
```

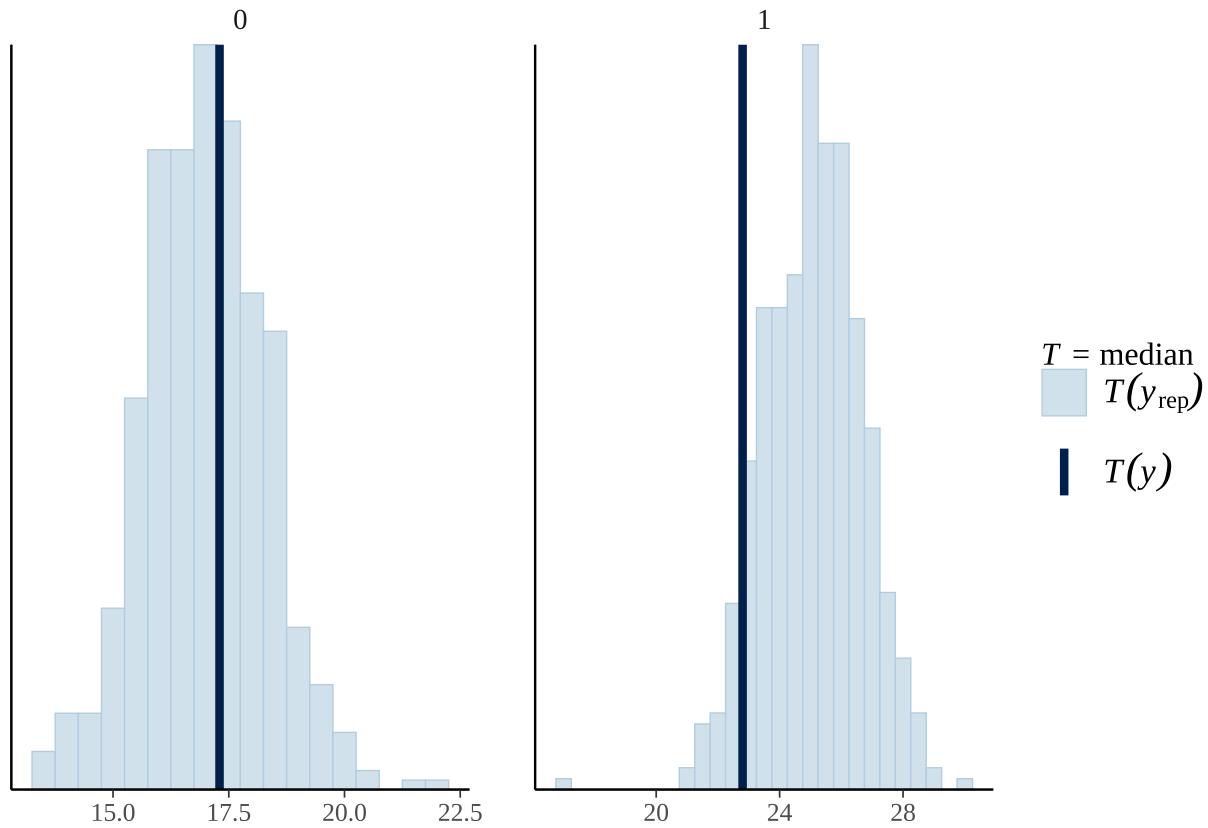


### 8.7.2 后验预测检查

用后验预测密度来检查模型如何覆盖你的数据。

```
ppd <- posterior_predict(model, draws=500)
ppd %>%
  ppc_intervals(y = mtcars$mpg, yrep = ., x = mtcars$wt, prob = 0.5) +
  labs(x = "Weight (1000 lbs)", y = "MPG", title = "")
```

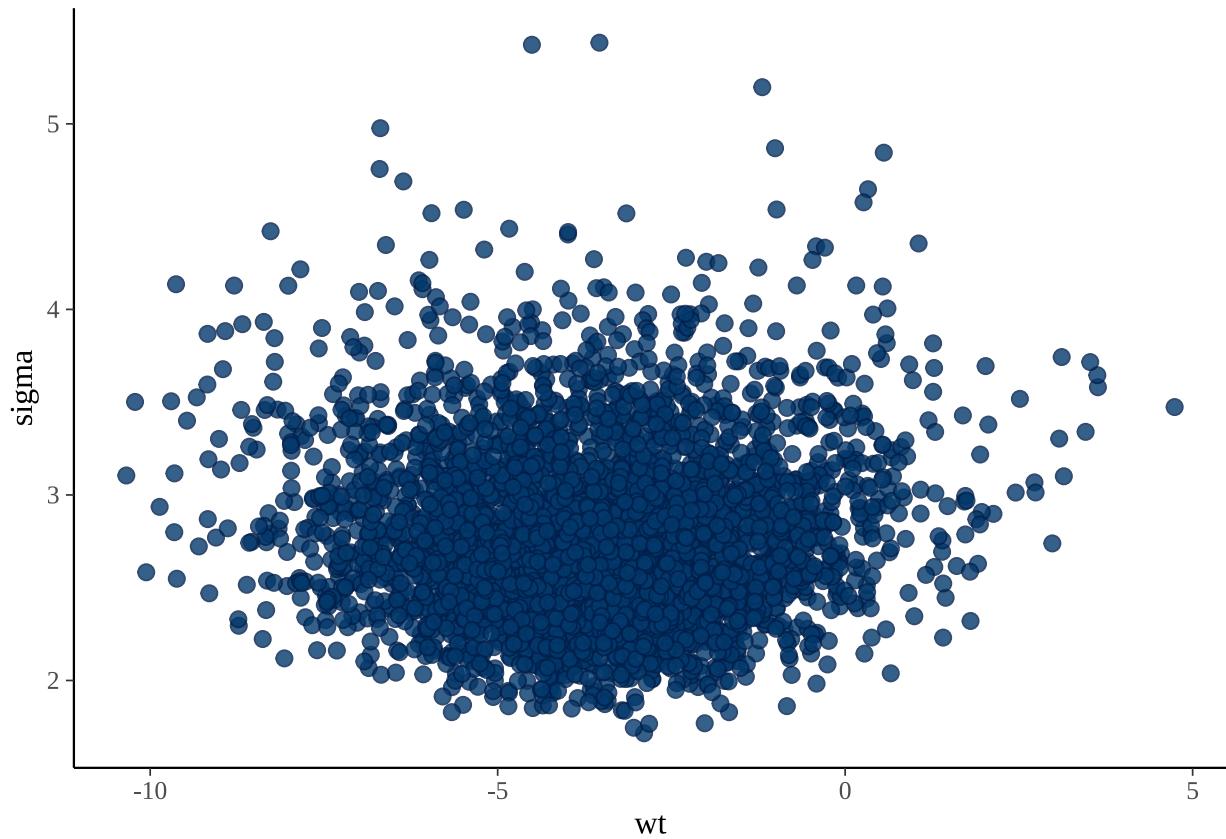




### 8.7.3 诊断

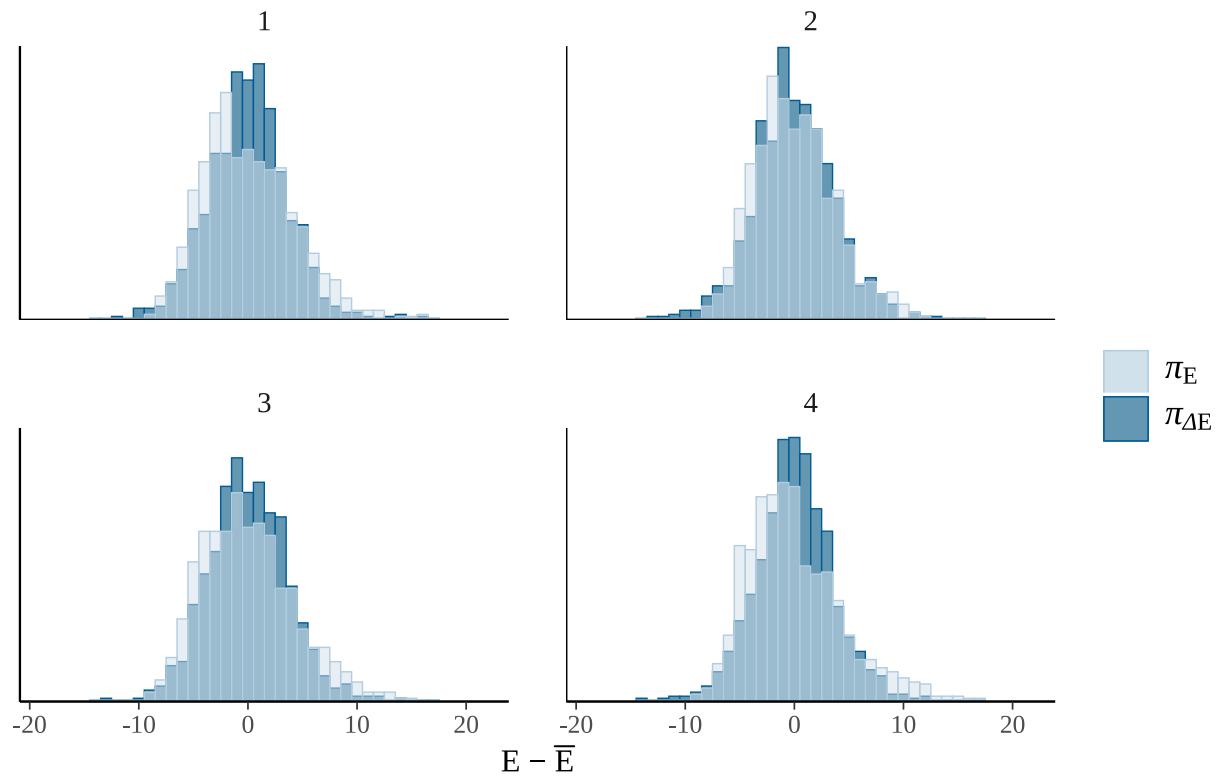
Bayesplot 可以方便地检查 rstanarm 默认使用的 NUTS 采样方法的诊断。

```
mcmc_scatter(posterior, pars = c("wt", "sigma"), np = nuts_params(model$stanfit))
```



```
np <- nuts_params(model$stanfit)
mcmc_nuts_energy(np, binwidth=1) +
  ggtitle("NUTS Energy Diagnostic")
```

## NUTS Energy Diagnostic





# Part III

## R 语言文档沟通



# Chapter 9

## R Markdown 入门教程

### 9.1 Rmarkdown 简介

Rmarkdown 是 R 语言环境中提供的 markdown 编辑工具，运用 rmarkdown 撰写文章，既可以像一般的 markdown 编辑器一样编辑文本，也可以在 rmarkdown 中插入代码块，并将代码运行结果输出在 markdown 里。R Markdown 格式，简称为 Rmd 格式，相应的源文件扩展名为.Rmd。输出格式可以是 HTML、docx、pdf、beamer 等。

介绍参考：李东风老师的《R 语言教程》。对于专注于用 R 语言写报告的数据分析师来说，rmarkdown 既提高了数据分析工作的便捷性，也提高了数据分析报告的复用性。markdown 的教程以及对应的编辑器介绍可见：R 沟通 | markdown 编辑器—Typora。

#### 9.1.1 官网视频介绍

先通过官方视频来看看 Rmarkdown 的介绍。

注：该视频来自官网，一般不一定打得出，所以小编为大家着想就下载下来了，以便大家更好地学习，可见这篇推送。视频已经非常清楚的介绍了 Rmarkdown 如何使用，内部构造、不同的输出类型，以及其他拓展（发布，与 github 相连）等。我们先对此进行简单了解即可，之后几期我会详细介绍。当然，官网也有一套 Rmarkdown 的入门教程，欢迎大家前去学习，官网截图如下：

R Markdown from R Studio

Get Started    Gallery    Formats    Articles    Book    References   

**Introduction**

- How It Works
- Code Chunks
- Inline Code
- Code Languages
- Parameters
- Tables
- Markdown Basics
- Output Formats
- Notebooks
- Slide Presentations
- Dashboards
- Websites
- Interactive Documents
- Cheatsheets

## Introduction

### Overview

R Markdown provides an authoring framework for data science. You can use a single R Markdown file to both

- save and execute code
- generate high quality reports that can be shared with an audience

R Markdown documents are fully reproducible and support dozens of static and dynamic output formats. This 1-minute video provides a quick tour of what's possible with R Markdown:

### Installation

Like the rest of R, R Markdown is free and open source. You can install the R Markdown package from CRAN with:

```
install.packages("rmarkdown")
```

### Get Started

其他参考资料可见这一期推文：R 分享 | Rmarkdown 参考资料分享和自制视频教程预告。关于 RMarkdown 可参考专著 (Xie, Allaire, and Grolemund 2019) 和 (Xie, Dervieux, and Riederer 2020)。RStudio 网站提供了一个 R Markdown 使用小抄的下载链接：( rmarkdown-2.0.pdf )[rmarkdown-2.0.pdf]。Pandoc 的文档见pandoc 网站，knitr 的详细文档参见网站knitr 文档。

## 9.2 Rmarkdown 流程演示

### 9.2.1 安装

假设你已经安装了 R (R Core Team 2020) 和 RStudio IDE。

不需要 RStudio，但建议使用 RStudio，因为它可使普通用户更轻松地使用 R Markdown。如果未安装 RStudio IDE，则必须安装 Pandoc，否则不需要单独安装 Pandoc，因为 RStudio 已将其捆绑在一起。接下来，Rstudio 中安装 rmarkdown 软件包，可以通过下面任意一种方式：

```
#Install from CRAN
install.packages('rmarkdown')

# Or if you want to test the development version,
#install from GitHub
if(!requireNamespace("devtools"))
install.packages('devtools')
devtools::install_github('rstudio/rmarkdown')
```

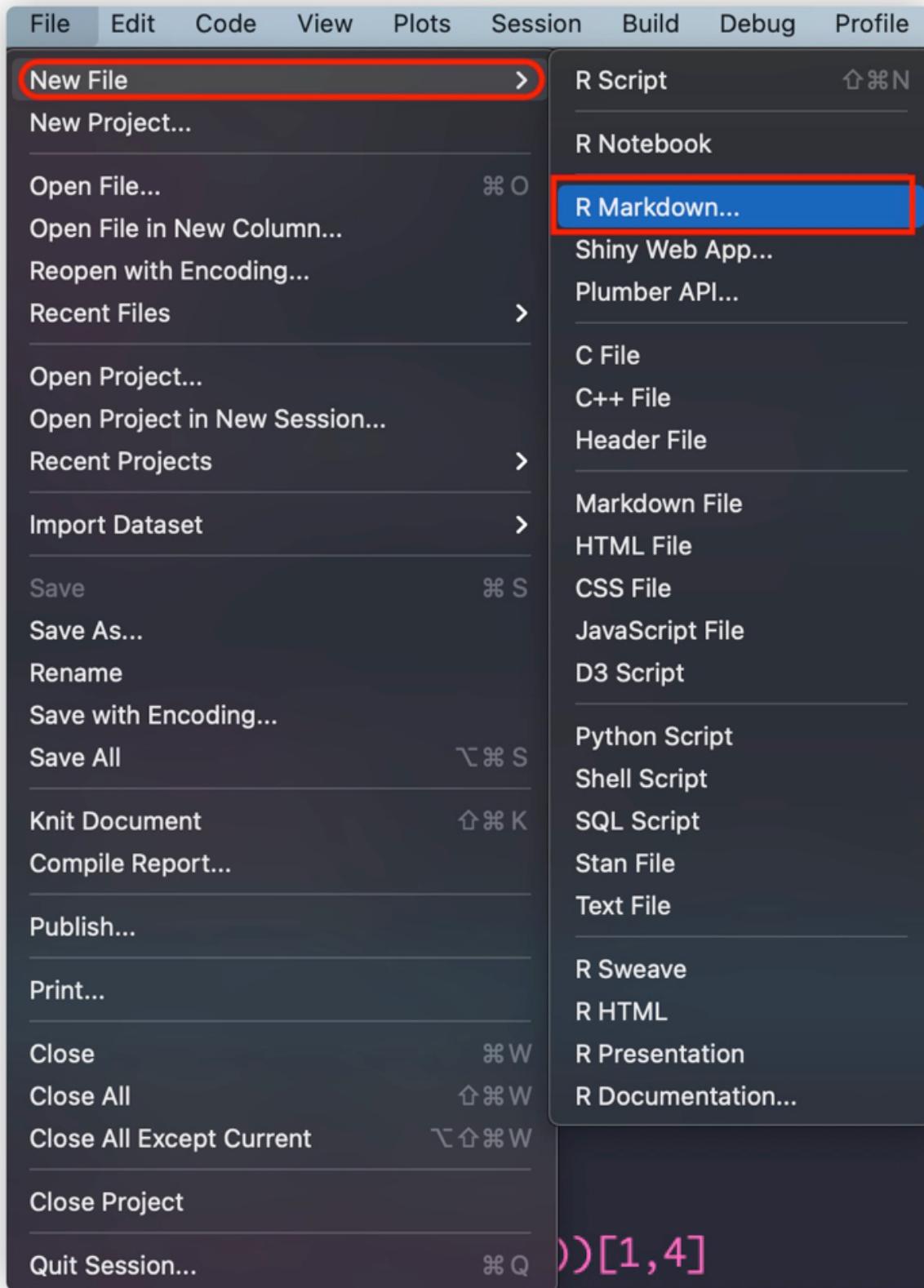
如果要生成 PDF 输出，则需要安装 LaTeX。对于从未安装过 LaTeX 的 R Markdown 用户，建议安装 TinyTeX：

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

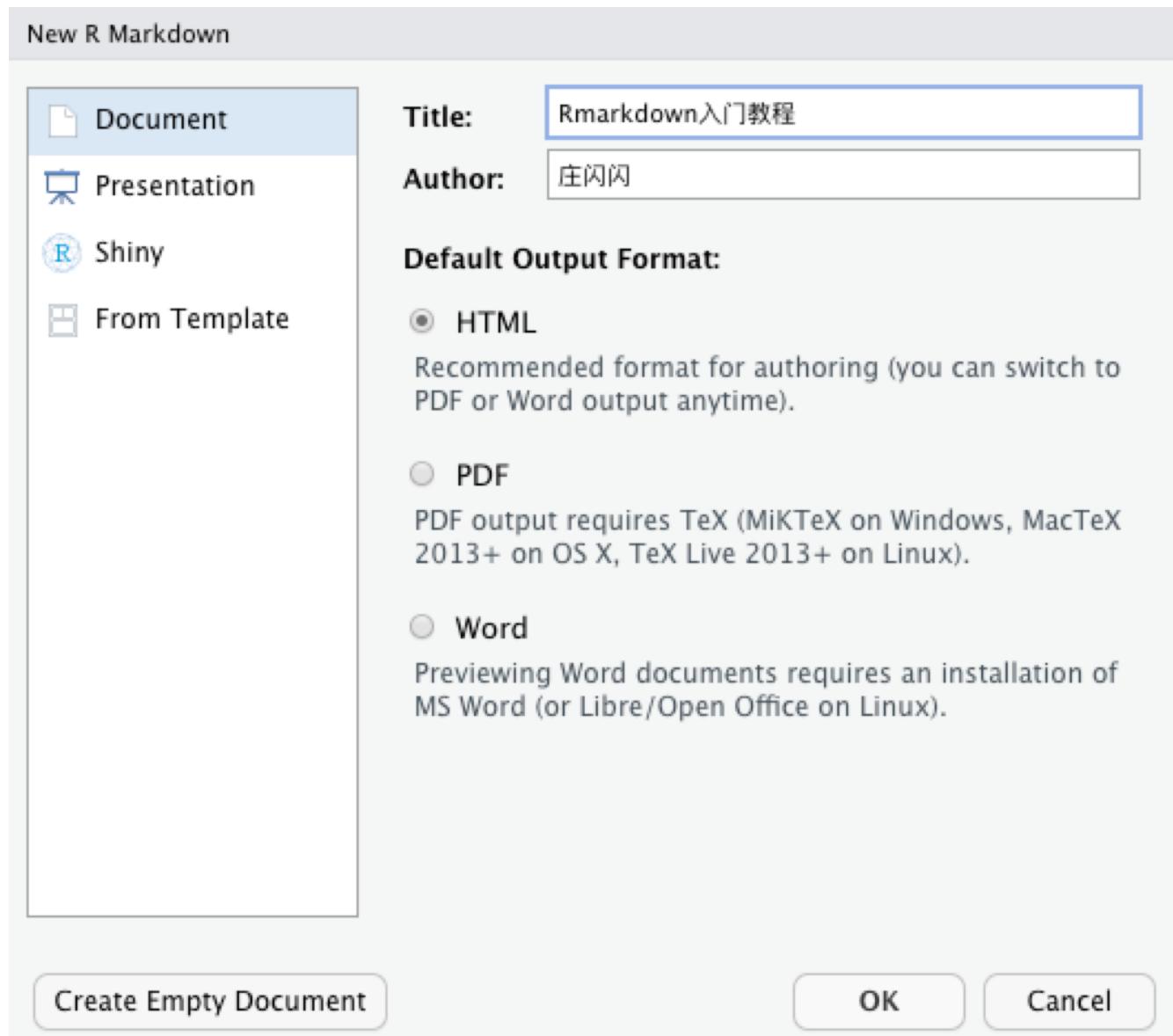
TinyTeX 是一种轻便，可移植，跨平台，易于维护的 LaTeX 发行版。R 配套软件包 `tinytex` 可以帮助你在将 LaTeX 或 R Markdown 文档编译为 PDF 时自动安装缺少的 LaTeX 软件包，并确保将 LaTeX 文档编译正确的次数以解决所有交叉引用。

注：如果 TinyTeX 通过上面代码无法正常安装，可以参考张敬信老师的知乎文章：搭建 Latex 环境：TinyTeX+RStudio 如果编译.rmd 格式时出现缺失某些 Latex 包，可以参考这节内容。### 2.2. 新建 Rmarkdown 项目

1. 点击 Rstudio 左上角的新建项目，选择 Rmarkdown 文件格式，即可建立一个 rmarkdown 编辑文件。



2. 在弹出的选项框里, 可以申明 rmarkdown 的 Title、Author 以及默认的输出文件格式, 一般可以选择 HTML、PDF、Word 格式, 具体见下图。



3. 在新建的 markdown 文件里, 主要包含三块内容: 1. YAML、2. markdown 文本、3. 代码块。

```

1---  

2 title: "Rmarkdown入门教程" 1  

3 author: "庄闪闪"  

4 date: "2/18/2021"  

5 output: html_document  

6---  

7  

8```{r setup, include=FALSE}  

9 knitr::opts_chunk$set(echo = TRUE)  

10```  

11  

12## R Markdown 3  

13  

14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML,  

  PDF, and MS Word documents. For more details on using R Markdown see  

  <http://rmarkdown.rstudio.com>.  

15  

16 When you click the **Knit** button a document will be generated that includes both content as  

  well as the output of any embedded R code chunks within the document. You can embed an R code  

  chunk like this:  

17

```

1) **YAML**: Rmarkdown 的头部文件 (上图 1 位置), YAML 定义了 rmarkdwon 的性质, 比如 title、author、date、指定 output 文件类型等。

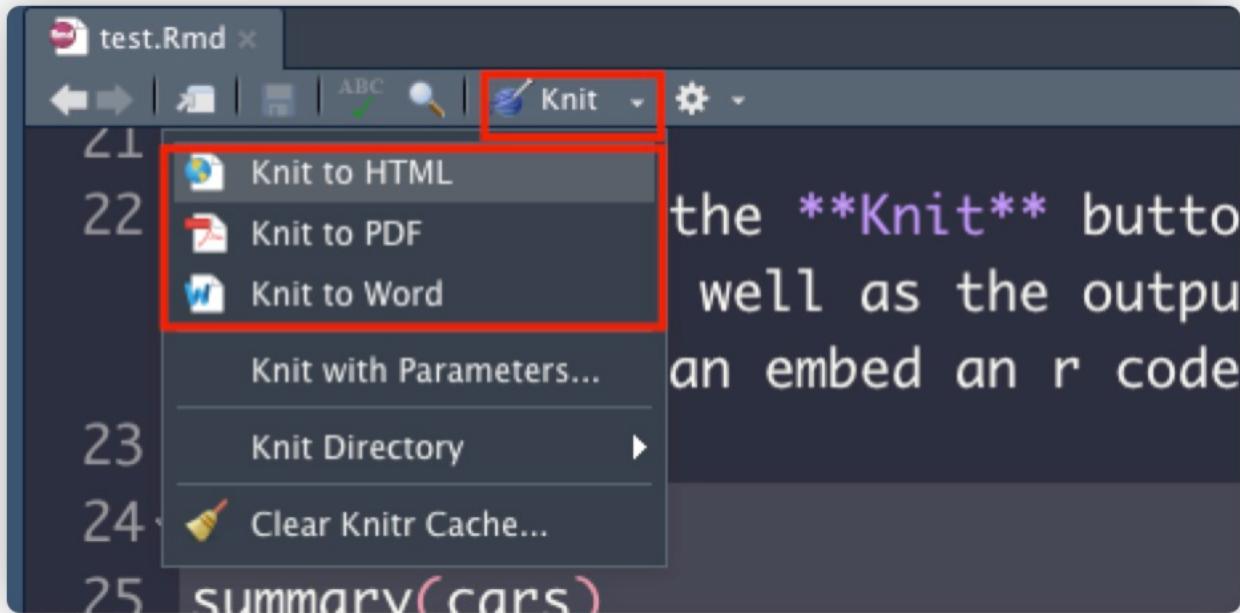
2) **markdown 文本**: rmarkdown 里的主要内容 (上图 3 位置), 由编辑人员按照 markdown 语法自行编写文本内容,

3) **代码块**: rmarkdown 的一个主要功能是可以执行文件内的代码块 (上图 2 位置), 并将代码执行结果展示在 markdown 里。这对撰写数据分析报告带来了极大的便利。Rmd 文件中除了 R 代码段以外, 还可以插入 Rcpp、Python、Julia、SQL 等许多编程语言的代码段, 常用编程语言还可以与 R 代码段进行信息交换。

这三个部分会在以后做详细介绍。## 2.3. Rmarkdown 的导出

rmarkdown 的导出方法有两种, 一种是依靠 Rstudio 手动导出, 另一种是基于命令行的导出方式。

### 9.2.1.1 手动导出



手动导出方法很简单,在完成 markdown 编辑后,手动点击上图红圈内 knit 按钮,选择导出格式类型即可,Rstudio 支持导出 PDF、html、word 三种类型。

### 9.2.1.2 命令行导出

命令行导出主要依靠 `rmarkdown::render` 实现, render 函数主要包含以下几个参数:

- **\*\*input\*\***: 指定需要导出的 rmarkdwon 文件地址
- **output\_format**: 指定需要导出的文件类型, 同样支持 pdf、word、html 等多种文件格式。若未指定 `output_format` 格式, 则输出 rmarkdown 文件中 `output` 指定的格式类型。

```
```{r}
rmarkdown::render("test.Rmd")
```
```

```

Console R Markdown ×
~/Desktop/wechat/庄闪闪原创/R/R分享/Rmarkdwon教程/Rmarkdown_teaching/ ↵
eak.lua --lua-filter /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rmarkdown/rmarkdown/lua/latex-div.lua --self-contained --standalone --section-divs --template /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rmarkdown/rmd/h/default.html --no-highlight --variable highlightjs=1 --variable 'theme:bootstrap' --include-in-header /var/folders/wt/t3n70vjs6g536860y9y_yr9r0000gn/T//RtmpW4LnL/rmarkdown-str314350350e2a.html --mathjax --variable 'mathjax-url:https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML'

Output created: test.html
> rmarkdown::render("test.Rmd")

```

## 9.2.2 Rstudio 界面介绍

若界面打开了 rmd 格式的文件时，RStudio 的界面发生了一些变化（多了一些按钮）。对界面做一些介绍，视频介绍在这。

注：演示的 Rstudio 版本为 1.4.1，老版本的界面可能有所不同。## 第三章：图、表和代码输出

## 9.2.3 代码输出

### 9.2.3.1 行内代码

1

有趣的案例：

1) 自动更新日期

```

date: "2022-07-23"
date: "23 July, 2022"
date: "Last compiled on 23 July, 2022"
```

```

具体细节可参考：<https://bookdown.org/yihui/rmarkdown-cookbook/update-date.html>

2) 文本结合代码

这里一共有 15 个人。

```
#### 代码块
```

```
```{r}
set.seed(1)
x <- round(rnorm(10), 2)
print(x)
```

```

```
```{r}
n = 10
dat = data.frame('group' = gl(n, 10), 'value' = rnorm(n * 10) + 10)
dat
dat1 = split(dat, dat$group)
```

```

R 代码块一般通过 {R} 来插入，插入代码段的快捷键：win[Ctrl+Alt+I] / mac[option + cmd + I]。

默认情况下代码和结果会在输出文件中呈现。如果通过参数来控制代码块运行结果的输出情况可以在 {r} 中设置。一般包括代码及运行结果的输出、图片表格格式定义等。关于代码段选项，详见 <https://yihui.name/knitr/options>。

这里小编给出一些常用的选项，文字版本较难理解的话，请配合我的 b 站视频教程。

### 1. eval 和 include 选项

加选项 `eval=FALSE`，可以使得代码仅显示而不实际运行。这样的代码段如果有标签，可以在后续代码段中被引用。

加选项 `include=FALSE`，则本代码段仅运行，但是代码和结果都不写入到生成的文档中。

### 2. echo 选项

`echo` 参数控制了 `markdown` 是否显示代码块。若 `echo=TRUE`，则表示代码块显示在 `markdown` 文档显示代码块；反之，代码块不出现在输出结果中。

```
```{r, echo = FALSE}
print(1:5)
```

```

结果为：

```
## [1] 1 2 3 4 5
```

3. collapse 选项一个代码块的代码、输出通常被分解为多个原样文本块中，如果一个代码块希望所有的代码、输出都写到同一个原样文本块中，加选项 `collapse=TRUE`。例如：

```
```{r, echo = FALSE}
sin(pi/2)
cos(pi/2)
...```

```

结果为：

```
sin(pi/2)
## [1] 1
cos(pi/2)
## [1] 6.123032e-17
```

代码和结果都在一个原样文本块中。

#### 4. prompt 和 comment 选项

`prompt=TRUE` 代码用 R 的大于号提示符开始。如果希望结果不用井号保护，使用选项 `comment=''`。

```
```{r prompt=TRUE, comment=''}
sum(1:5)
...```

```

结果为：

```
> sum(1:5)
[1] 15
```

#### 5. results 选项

用选项 `results=` 选择文本型结果的类型。取值有：

- `markup`，这是缺省选项，会把文本型结果变成HTML的原样文本格式。

- `hide`，运行了代码后不显示运行结果。

- `hold`，一个代码块所有的代码都显示完，才显示所有结果。

- `asis`，文本型输出直接进入到 HTML 文件中，这需要 R 代码直接生成HTML标签，`knitr` 包的 `kable()` 函数。

例如，`results='hold'` 的示例：

```
```{r collapse=TRUE, results='hold'}
sin(pi/2)
cos(pi/2)
```

```

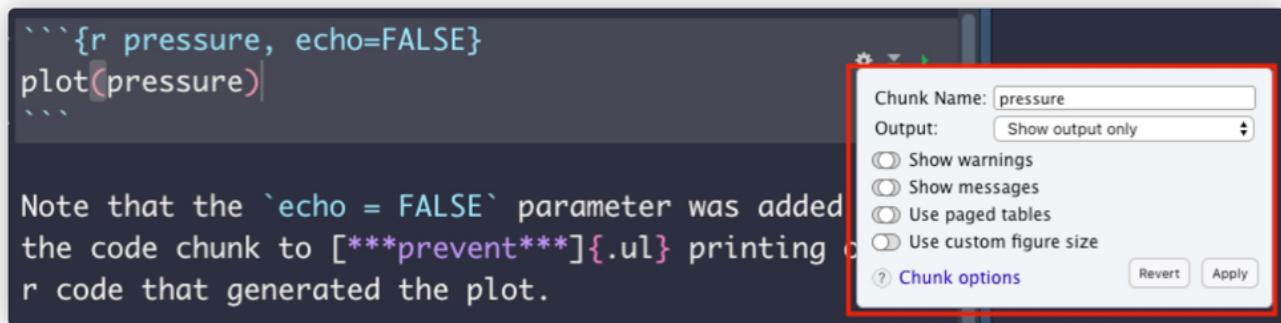
结果为:

```
sin(pi/2)
cos(pi/2)
## [1] 1
## [1] 6.123032e-17
```

## 6. 错误信息选项

- `warning=FALSE` 使得代码段的警告信息不进入编译结果，而是在控制台（console）中显示。有一些扩展包的载入警告可以用这种办法屏蔽。
- `error=FALSE` 可以使得错误信息不进入编译结果，而是出错停止并将错误信息在控制台中显示。
- `message=FALSE` 可以使得 message 级别的信息不进入编译结果，而是在控制台中显示。

还有关于图片的设置，这个我们放到下面来说当然你也可以通过 Rstudio 界面进行部分参数的设置（更加便捷）：



具体演示可见 b 站视频。

## 7. 全局设置

若 markdown 内的代码块存在一样的参数设置，则可以提前设计好全局的代码块参数。全局代码块通过 `knitr::opts_chunk$set` 函数进行设置，一般设置在 YAML 文件下方，具体见下图。

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```

### 9.2.4 图片输出

### 9.2.4.1 插入 R 代码生成的图形

```
```{r, fig.height = 8,fig.width = 6}
plot(1:10)
```

1. `fig.show`: 设置了图片输出方式

- ‘fig.show=‘asis’’: 表示 plot 在产生他们的代码后面
- ‘fig.show=‘hold’’: 所有代码产生的图片都放在一个完整的代码块之后
- ‘fig.show=‘animate’’: 表示将所有生成的图片合成一个动画图片

可以参考: <https://bookdown.org/yihui/rmarkdown-cookbook/animation.html>

- ‘fig.show=‘hide’’: 表示产生所有图片, 但是并不展示

```
'markdown{r, fig.show='animate'}
for (i in 1:2) {
  pie(c(i %% 2, 6), col = c('red', 'yellow'), labels = NA)
}
```

2. `fig.width`: 设置图片输出的宽度

3. `fig.height`: 设置图片输出的高度

4. `fig.align`: 设置图片位置排版格式, 默认为 “left”, 可以为 “right” 或者 “center”。

5. `fig.cap`: 设置图片的标题

6. `fig.subcap`: 设置图片的副标题

7. `out.width` 和 `out.height` 选项指定在输出中实际显示的宽和高, 如果使用如 “90%” 这样的百分数单位则可以自动适应输出的大小。

### 9.2.4.2 插入外部图形文件

如果一个图不是由一个 R 代码块生成的, 你可以用两种方式包含它:

- 方法一

使用 Markdown 语法 `![caption](path/to/image)`, 你可以使用 `width` 和 `height` 属性来设置图像的大小, 例如:

```
![图的标题](xxx.png){width=50%}
```

注意: 图片文件放的位置 (如果和 rmd 同一目录, 则可以直接 xxx.png; 如果在其他位置记得加上相对路径)。

```

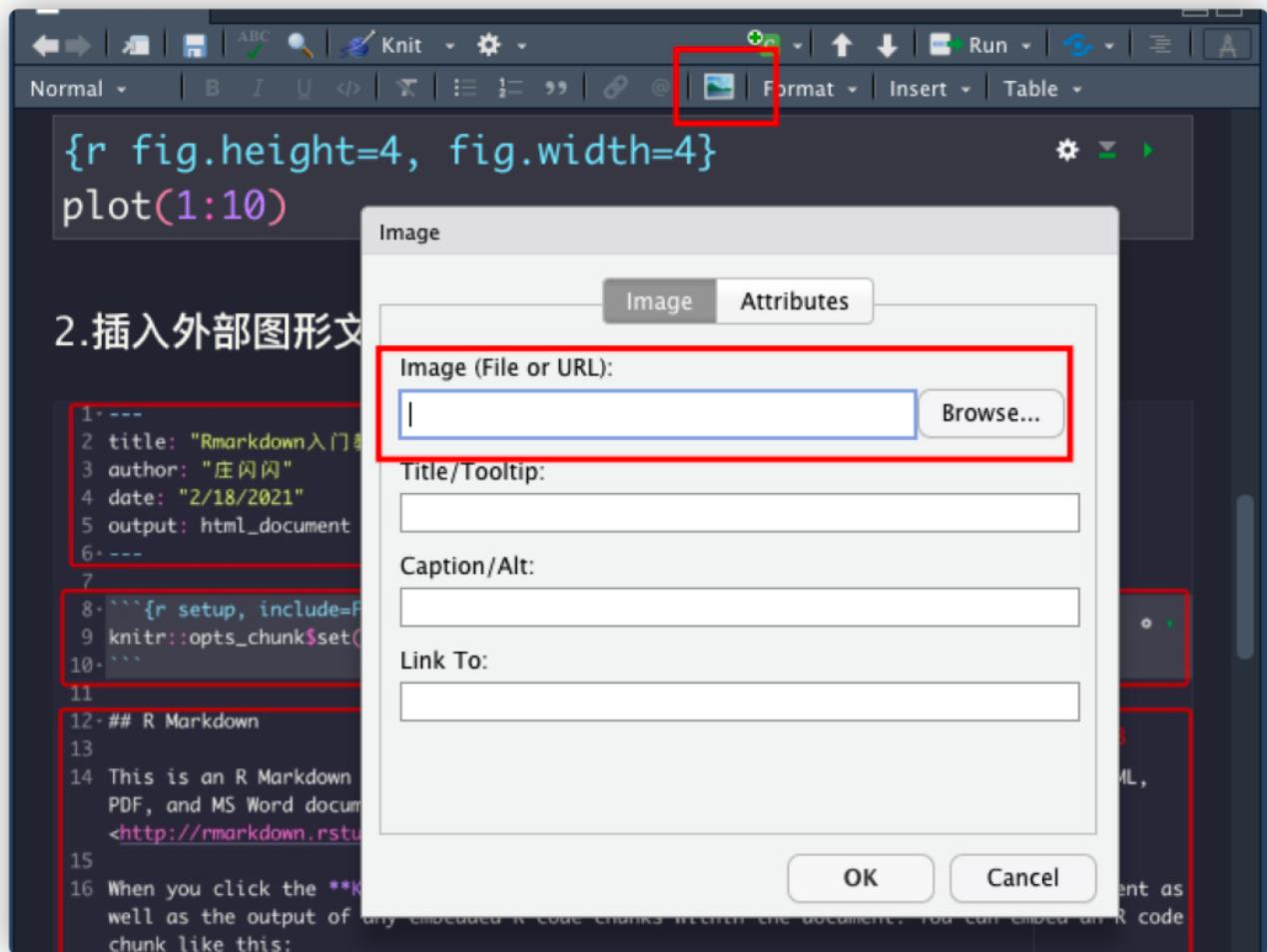

1:---
2 title: "Rmarkdown入门教程"          1
3 author: "庄闪闪"
4 date: "2/18/2021"
5 output: html_document
6:---

7
8:```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10:```

11
12:## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML,
15 PDF, and MS Word documents. For more details on using R Markdown see
16 <http://rmarkdown.rstudio.com>.
17
```

- 方式二

在 source editor 情况下，直接外部拉入图形即可，会自动保存在相对文件夹的 images 中，或者点击图形按钮导入。



- 方式三

在代码块中使用 knitr 函数 `knitr::include_graphics()`。图片尺寸更改与插入 R 代码生成的图形的情况相同。

```
```{r, echo=FALSE , out.width="50%"}  
knitr::include_graphics("xxx.png")  
```
```

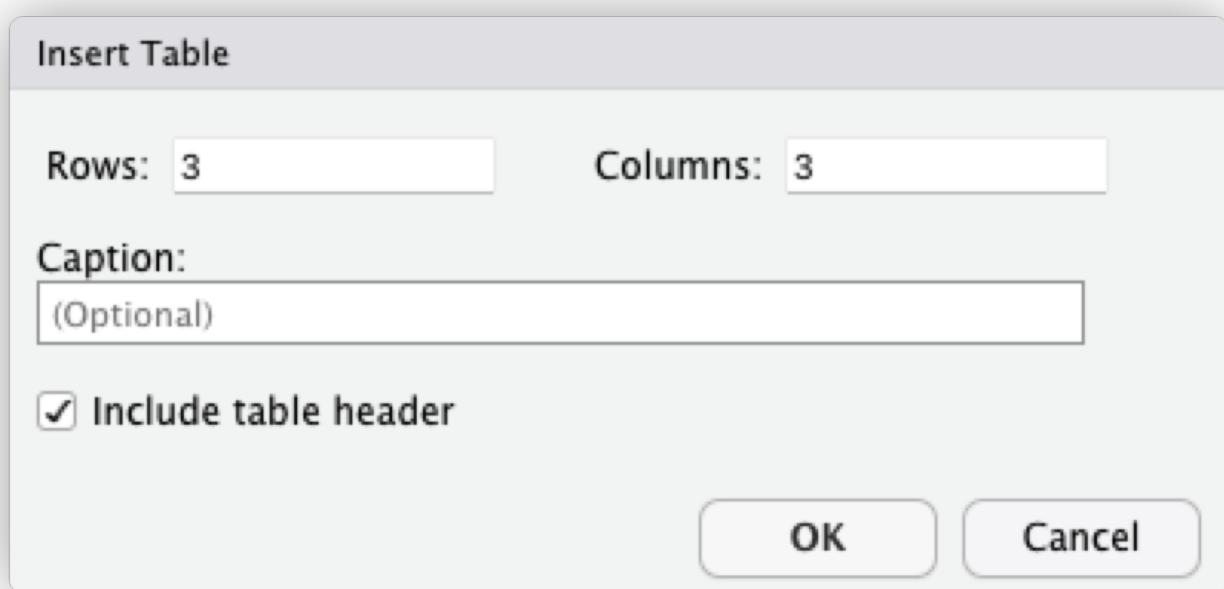
## 9.2.5 表格输出

### 9.2.5.1 外部表格输入

- markdown 格式

| x | y | value |
|---|---|-------|
| 1 | 2 |       |
| 1 | 2 |       |
| 1 | 2 |       |

- Typora 格式



### 9.2.5.2 内部代码输出的表格

例子：计算线性回归后，`summary()` 函数的输出中有 `coefficients` 一项，是一个矩阵，如果直接文本显示比较难看：

```
x <- 1:10; y <- x^2; lmr <- lm(y ~ x)
co <- summary(lmr)$coefficients
print(co)
```

- `knitr` 包的 `kable()`

`knitr` 包提供了一个 `kable()` 函数可以用来把数据框或矩阵转化成有格式的表格，支持 HTML、docx、LaTeX 等格式。

可以用 `knitr` 包的 `kable` 函数来显示：

```
knitr::kable(co)
```

|             | Estimate | Std. Error | t value   | Pr(> t )  |
|-------------|----------|------------|-----------|-----------|
| (Intercept) | -22      | 5.5497748  | -3.964125 | 0.0041530 |
| x           | 11       | 0.8944272  | 12.298374 | 0.0000018 |

`kable()` 函数的 `digits=` 选项可以控制小数点后数位数，`caption=` 选项可以指定表的标题内容。

- `pander` 包的 `pander` 函数

其 `pander()` 函数可以将多种 R 输出格式转换成 `knitr` 需要的表格形式。如

```
pander::pander(lmr)
```

| Fitting linear model: y ~ x |          |            |         |           |
|-----------------------------|----------|------------|---------|-----------|
|                             | Estimate | Std. Error | t value | Pr(> t )  |
| (Intercept)                 | -22      | 5.55       | -3.964  | 0.004153  |
| x                           | 11       | 0.8944     | 12.3    | 1.778e-06 |

但是，经过试验发现，表中中有中文时 `pander` 包会出错，这里参考公众号R 友舍。

- 其他包

`tables` (Murdoch 2020), `pander` (Darócz and Tsegelskyi 2018), `tangram` (Garbett 2020), `ztable` (Moon 2020)，和 `condformat` (Oller Moreno 2020) 等。

### 9.2.6 表格渲染

通过前面可以看到：用 `knitr::kable()` 输出表格结果其实不是非常美观，并且很多功能都不能实现。这时我们可以用 `kableExtra`(Zhu 2020)、`huxtable` (Hugh-Jones 2020) 等扩展包来美化表格。

其他拓展包可以参考：<https://bookdown.org/yihui/rmarkdown-cookbook/table-other.html> - `kableExtra` 包

本文以 `kableExtra` 包为例，介绍 `rmarkdown` 中渲染表格的相关函数。它在 <https://haozhu233.github.io/kableExtra/> 上有大量文档，其中提供了许多示例，说明如何针对 HTML 或 LaTeX 输出自定义 `kable()` 输出。建议阅读其文档，本节中仅介绍一些示例介绍。

`kableExtra` 包可以使用管道符号 `%>%` 操作，例如

```
library(knitr)
library(kableExtra)
kable(iris) %>%
  kable_styling(latex_options = "striped")
```

#### 9.2.6.1 安装

```
# install from CRAN
install.packages("kableExtra")
# install the development version
remotes::install_github("haozhu233/kableExtra")
```

#### 9.2.6.2 表格外框设置

`bootstrap_options = "bordered"` 构建有边框的表格，其他可调节的名称可通过帮助文档获取。

```
x_html <- knitr:: kable(head(rock), "html")
kableExtra::kable_styling(x_html,bootstrap_options = "bordered")
```

#### 9.2.6.3 设置表格的宽度

使用 `full_width = F` 使得表格横向不会填满整个页面，默认情况下 `full_width = T`。

```
x_html <- knitr:: kable(head(rock), "html")
kableExtra::kable_styling(x_html,bootstrap_options = "striped",
                          full_width = F)
```

| area | peri    | shape     | perm |
|------|---------|-----------|------|
| 4990 | 2791.90 | 0.0903296 | 6.3  |
| 7002 | 3892.60 | 0.1486220 | 6.3  |
| 7558 | 3930.66 | 0.1833120 | 6.3  |
| 7352 | 3869.32 | 0.1170630 | 6.3  |
| 7943 | 3948.54 | 0.1224170 | 17.1 |
| 7979 | 4010.15 | 0.1670450 | 17.1 |

注意：上面例子 `knitr:: kable` 制定了 `kable` 函数来自 `knitr` 包，目的是方式和其他包内同名函数冲突。

另一种写法，如果想使用管道函数，需要加载 `kableExtra`。其他代码也类似，大家要学会举一反三噢！

```
library(knitr)
library(kableExtra)
kable(head(rock), "html") %>%
```

```
kable_styling(x_html,bootstrap_options = "striped",
               full_width = F)
```

#### 9.2.6.4 设置表格的对齐方式

使用 `position = "left"` 使得整个表格像左对齐，当然也可以中心对齐和右对齐，视情况而定。

```
x_html <- knitr:::kable(head(rock), "html")
kableExtra::kable_styling(x_html,bootstrap_options = "striped",
                           full_width = F,
                           position = "left")
```

| area | peri    | shape     | perm |
|------|---------|-----------|------|
| 4990 | 2791.90 | 0.0903296 | 6.3  |
| 7002 | 3892.60 | 0.1486220 | 6.3  |
| 7558 | 3930.66 | 0.1833120 | 6.3  |
| 7352 | 3869.32 | 0.1170630 | 6.3  |
| 7943 | 3948.54 | 0.1224170 | 17.1 |
| 7979 | 4010.15 | 0.1670450 | 17.1 |

#### 9.2.6.5 设置表格的字体大小

使用 `font_size = 20` 可以将字体大小改为 20。

```
x_html <- knitr:::kable(head(rock), "html")
kableExtra::kable_styling(x_html,bootstrap_options = "striped",
                           full_width = T,
                           font_size = 20)
```

| area | peri    | shape     | perm |
|------|---------|-----------|------|
| 4990 | 2791.90 | 0.0903296 | 6.3  |
| 7002 | 3892.60 | 0.1486220 | 6.3  |
| 7558 | 3930.66 | 0.1833120 | 6.3  |
| 7352 | 3869.32 | 0.1170630 | 6.3  |
| 7943 | 3948.54 | 0.1224170 | 17.1 |
| 7979 | 4010.15 | 0.1670450 | 17.1 |

### 9.2.6.6 设置表格的行与列

这里使用的函数是 `column_spec()`，其中以下代码含义为：制定前两列数据，字体加粗、颜色为白色，表格填充为 "#D7261E"。而行的设置与列类似，使用函数名为 `row_spec()`。

```
x_html <- knitr:: kable(head(rock), "html")
x_html <- kableExtra::kable_styling(x_html,
                                      bootstrap_options = "striped",
                                      full_width = T)

kableExtra::column_spec(x_html, 1:2,
                       bold = T,
                       color = "white",
                       background = "#D7261E")
```

| area | peri    | shape     | perm |
|------|---------|-----------|------|
| 4990 | 2791.90 | 0.0903296 | 6.3  |
| 7002 | 3892.60 | 0.1486220 | 6.3  |
| 7558 | 3930.66 | 0.1833120 | 6.3  |
| 7352 | 3869.32 | 0.1170630 | 6.3  |
| 7943 | 3948.54 | 0.1224170 | 17.1 |
| 7979 | 4010.15 | 0.1670450 | 17.1 |

```
x_html <- knitr:: kable(head(rock), "html")
x_html <- kableExtra::kable_styling(x_html,
                                      bootstrap_options = "striped",
                                      full_width = T)

kableExtra::row_spec(x_html, 1:2,
                      bold = T,
                      color = "white",
                      background = "#D7261E")
```

| area | peri    | shape     | perm |
|------|---------|-----------|------|
| 4990 | 2791.90 | 0.0903296 | 6.3  |
| 7002 | 3892.60 | 0.1486220 | 6.3  |
| 7558 | 3930.66 | 0.1833120 | 6.3  |
| 7352 | 3869.32 | 0.1170630 | 6.3  |
| 7943 | 3948.54 | 0.1224170 | 17.1 |
| 7979 | 4010.15 | 0.1670450 | 17.1 |

### 9.2.6.7 其它表格渲染

这里给出一个有趣的例子，具体内部不做过多解释，b 站视频给出解释。文字真的很难说明。

```
library(kableExtra)
data = plyr::mutate(rock[1:10, ],
                    perm = cell_spec(perm,"html",
                                      color = "white",
                                      bold = T,
                                      background = spec_color(1:10,
                                                               end = 0.9,
                                                               option = "A",
                                                               direction = -1)),
                    shape = ifelse(shape > 0.15,
                                   cell_spec(shape,
                                             "html",
                                             color = "white",
                                             background = "#D7261E",
                                             bold = T),
                                   cell_spec(shape, "html",
                                             color = "green",
                                             bold = T)))
x_html <-knitr::kable(data,"html", escape = F, align = "c")
x_html <-row_spec(x_html,0, color = "white", background = "#696969" )
kable_styling(x_html,"striped")
```

| area | peri    | shape     | perm |
|------|---------|-----------|------|
| 4990 | 2791.90 | 0.0903296 | 6.3  |
| 7002 | 3892.60 | 0.148622  | 6.3  |
| 7558 | 3930.66 | 0.183312  | 6.3  |
| 7352 | 3869.32 | 0.117063  | 6.3  |
| 7943 | 3948.54 | 0.122417  | 17.1 |
| 7979 | 4010.15 | 0.167045  | 17.1 |
| 9333 | 4345.75 | 0.189651  | 17.1 |
| 8209 | 4344.75 | 0.164127  | 17.1 |
| 8393 | 3682.04 | 0.203654  | 119  |
| 6425 | 3098.65 | 0.162394  | 119  |

### 9.3 Rmarkdown 的主题格式

Rmarkdown 作为可复用报告的优秀工具，除了提供文档编辑、图表输出外，还有许多主题格式供使用者选择。除了默认的主题外，还可以通过加载 `rticles`、`prettydoc`、`rmdformats`、`tufte` 等包获取更多主题格式。下面我们看看几类扩展包里的主题样式。

注：接下来的教程我是已经安装这些包了，没安装的记得提前安装好！不然运行会出错。### 4.1.

`rticles` 包

官网：<https://github.com/rstudio/rticles>；

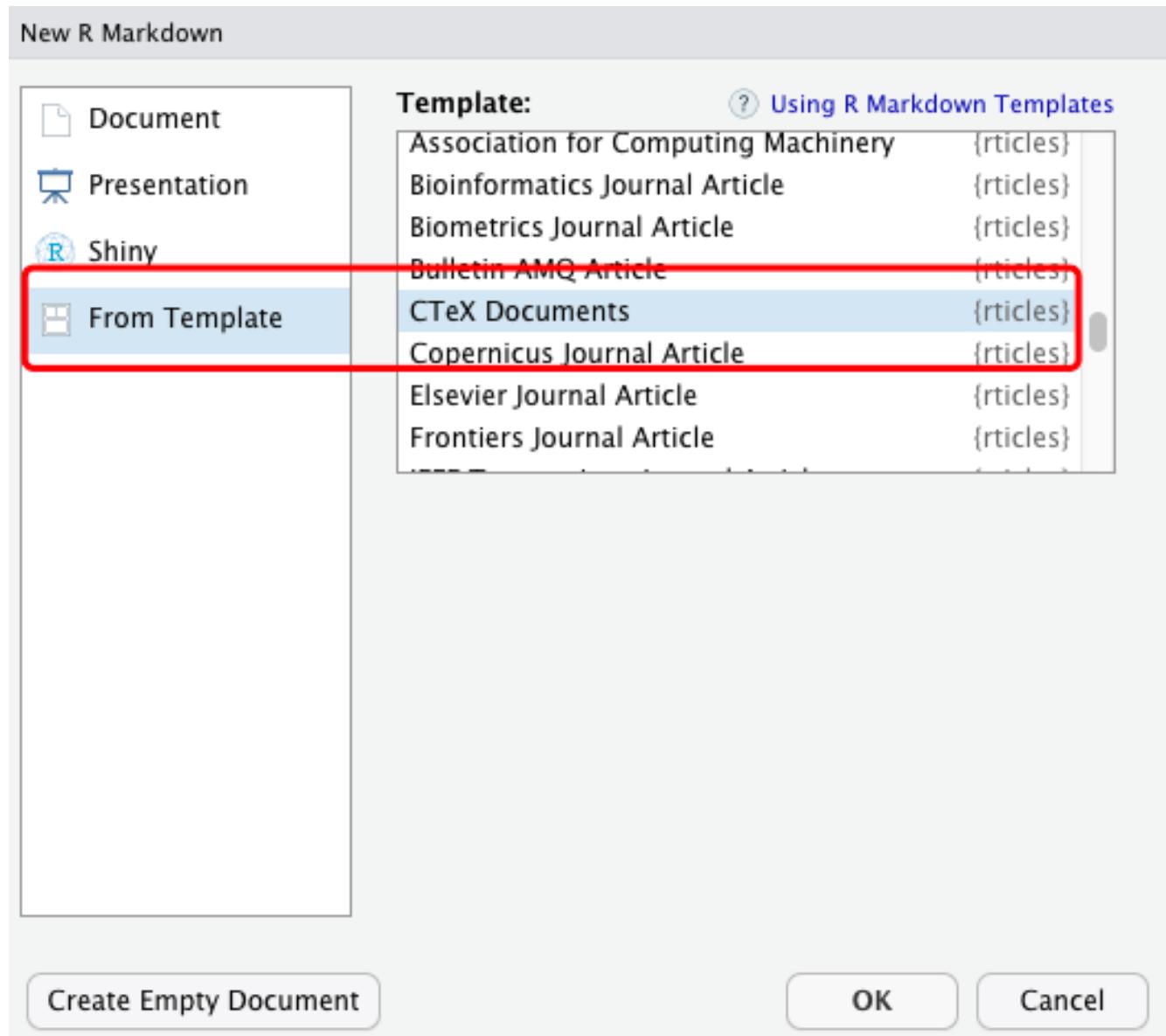
具体教程：<https://bookdown.org/yihui/rmarkdown/rticles-templates.html>

`rticles` 软件包提供了各种期刊和出版商的模板：

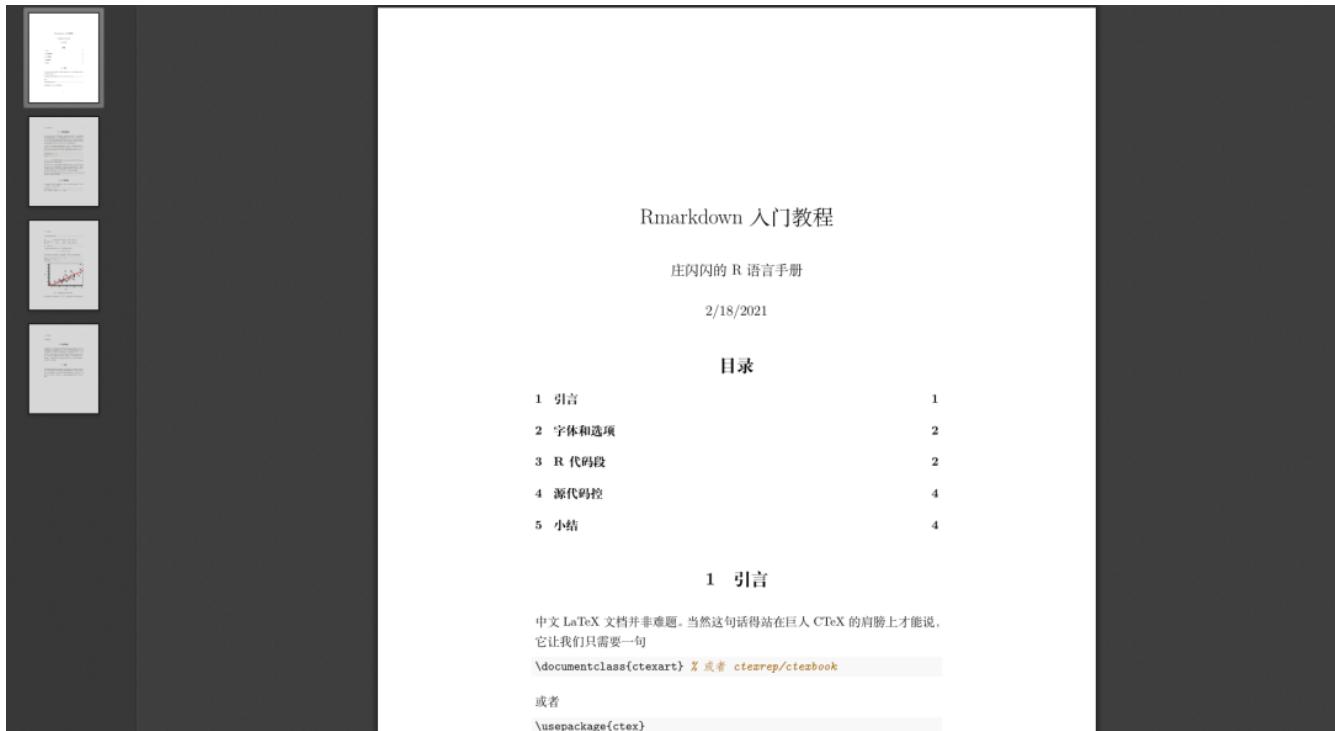
- JSS articles (Journal of Statistical Software)
- R Journal articles
- CTeX documents(中文 pdf, 强烈推荐! )
- ACM articles (Association of Computing Machinery)
- ACS articles (American Chemical Society)
- AMS articles (American Meteorological Society)
- PeerJ articles
- Elsevier journal submissions
- AEA journal submissions (American Meteorological Society)
- IEEE Transaction journal submissions
- Statistics in Medicine journal submissions
- Royal Society Open Science journal submissions
- Bulletin de l'AMQ journal submissions
- MDPI journal submissions
- Springer journal submissions

在此只对下面一个模板进行演示，其他模板操作类似，但是一般模板是不能包含中文字体的哦！- CTeX Documents (中文版本)

下载完对应的包之后，找到对应模板打开即可。输出 pdf 是需要配置 tex 环境的哦！建议安装 Tinytex，具体安装教程见前面。



编译后得到的结果，这是他模板原始的样子，如果想调整页面行间距，字体颜色等，请见下次推文（公众号：庄闪闪的 R 语言手册），我将会把这些小细节进行汇总。



### 9.3.1 rmdformats 包

之后的这几个包，内部都包含了挺多模板的，下面 yaml 文件只是其中一个，如果想尝试该包内部其他模板，请根据上述操作进行，选择好模板，编译之后看看是不是你想要的模板。

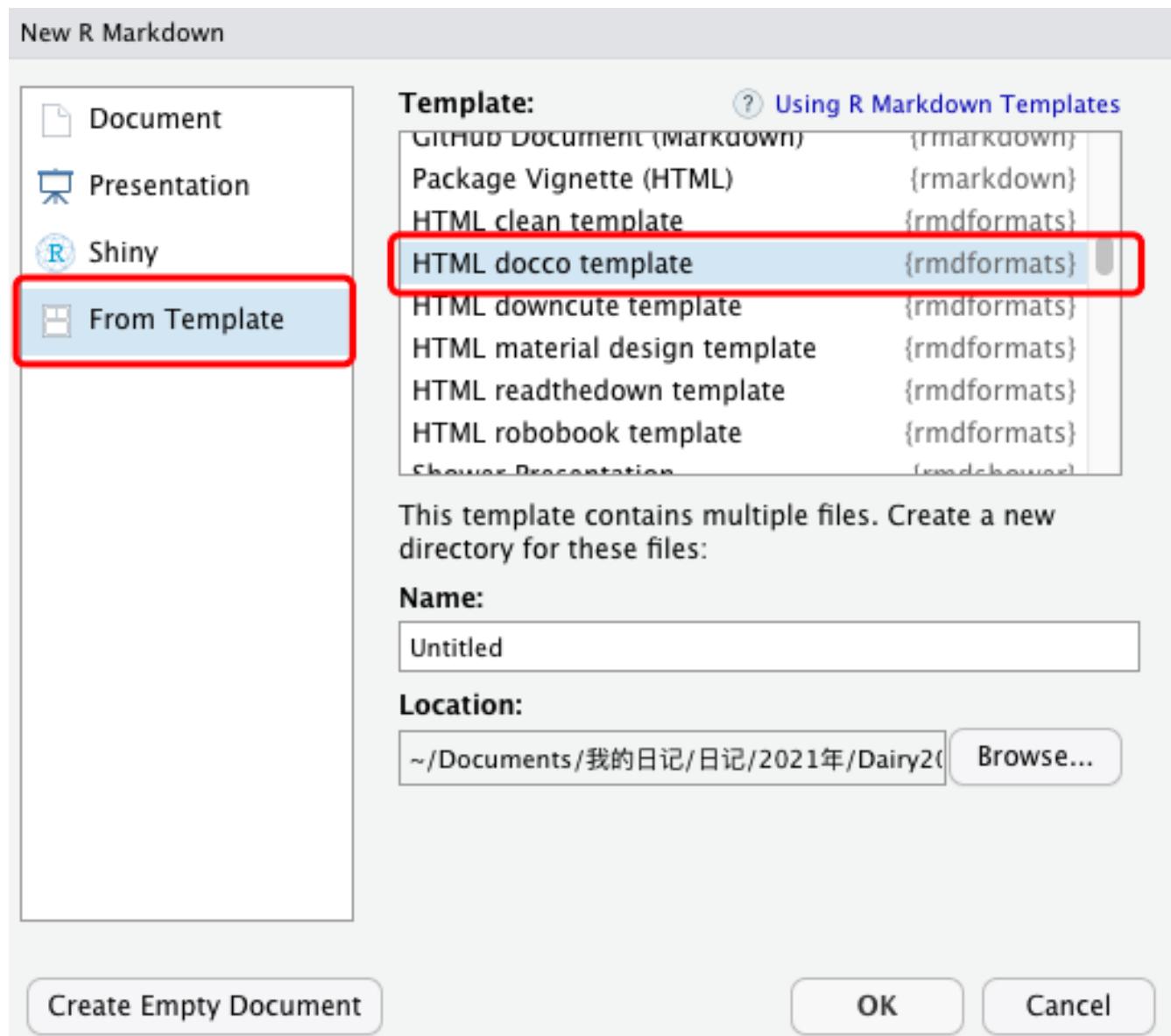
接下来我对部分相对不错的模板进行展示，你可以直接复制我的头部文件到.rmd 格式中，或者打开模板窗口进行选择（操作在下面）。- 方法一

```
---
title: "Rmarkdown 入门教程"
author: "庄闪闪的 R 语言手册"
date: "2/18/2021"
output:
  rmdformats::readthedown:
    self_contained: true
    thumbnails: true
    lightbox: true
    gallery: false
    highlight: tango
---
```

The screenshot shows a web browser displaying an R Markdown document. The title bar says 'Untitled.html | Open in Browser | Find'. The main content area has a red header bar with the text 'Rmarkdown入门教程'. On the left, there's a sidebar with 'R Markdown' and 'Including Plots' options. The main content starts with a section titled 'R Markdown' which contains a note about its purpose and a link to the official site. Below it is a code chunk with the command 'summary(cars)'. The output of this chunk is a table showing the distribution of 'speed' and 'dist' variables. Following this is a section titled 'Including Plots' with a note about embedding plots. A scatter plot of 'dist' vs 'speed' is shown, generated by the R code 'plot(cars)'. At the bottom left, there's a footer with a user icon, the name '庄闪闪', and the date '2/18/2021'.

## 方法二：

在你安装完该包之后你可以使用通过按钮新建该模版（其实他有很多类似的模板，我这里只展现了一种）：



以下模板也可以通过这种方式构建，前提是你安装了这个包，这样你就可以在 From Template 中找到该包对应的模板了。### 4.3. prettydoc 包

```
---
```

```
title: "Rmarkdown 入门教程"
author: "庄闪闪的 R 语言手册"
date: "2/18/2021"
output:
  prettydoc::html_pretty:
    theme: cayman
    highlight: github
```



## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed         dist
## Min.   :4.0   Min.   : 2.00
## 1st Qu.:12.0  1st Qu.:26.00
## Median :15.0  Median :36.00
## Mean   :15.4  Mean   :42.98
## 3rd Qu.:19.0  3rd Qu.:56.00
## Max.   :25.0  Max.   :120.00
```

## Including Plots

You can also embed plots, for example:



---

```
title: "Rmarkdown 入門教程"
author: "庄闪闪的 R 语言手册"
date: "2/18/2021"
output:
  tufte::tufte_html: default
---
```

Untitled.html | Open in Browser | Find | Publish |

# Rmarkdown入门教程

庄闪闪的R语言手册  
2/18/2021

*R Markdown*

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
##      speed      dist
## Min.   :4.0   Min.   : 2.00
## 1st Qu.:12.0  1st Qu.:26.00
## Median :15.0  Median :36.00
## Mean   :15.4  Mean   :42.98
## 3rd Qu.:19.0  3rd Qu.:56.00
## Max.   :25.0  Max.   :120.00
```

*Including Plots*

You can also embed plots, for example:

```
---
title: "Rmarkdown 入门教程"
author: "庄闪闪的 R 语言手册"
date: "2/18/2021"
output:
  html_document:
    theme: cerulean
    highlight: tango
---
```

## Rmarkdown入门教程

庄闪闪的R语言手册

2/18/2021

### R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

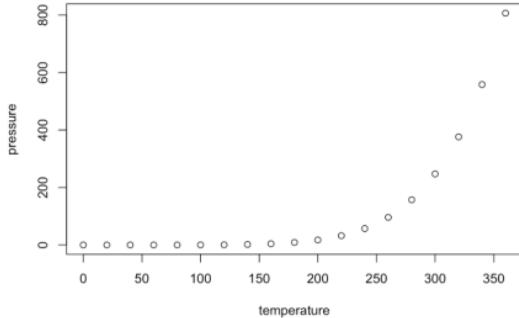
When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##   speed      dist
## Min. : 4.0  Min. :  2.00
## 1st Qu.:12.0  1st Qu.: 26.00
## Median :15.0  Median : 36.00
## Mean   :15.4  Mean   : 42.98
## 3rd Qu.:19.0  3rd Qu.: 56.00
## Max.  :25.0  Max.  :120.00
```

### Including Plots

You can also embed plots, for example:



## 9.4 总结一些常用技巧

关于 RMarkdown 使用时，小编日常会使用的一些有用技巧，当然我也是通过学习谢大大的Rmarkdown-cookbook以及日常使用需求上网搜的解决方案，在此分享给大家。如果大家还有其他什么需求，可以在留言板留言。或者有其他实用技巧也欢迎分享！

### 9.4.1 修改某些字体颜色

Markdown 语法没有用于更改文本颜色的内置方法。我们可以使用 HTML 和 LaTeX 语法来更改单词的格式

- 对于 HTML，我们可以将文本包装在标记中，并使用 CSS 设置颜色，例如 `<span style = " color: red; > text </ span>`。
- 对于 PDF，我们可以使用 LaTeX 命令 `\textcolor{red}{text}`。这需要使用 LaTeX 软件包 xcolor，该软件包已包含在 Pandoc 的默认 LaTeX 模板中。

作为更改 PDF 文本颜色的示例：

我是\textcolor{blue}{庄闪闪} 呀！欢迎关注我的\textcolor{red}{公众号}：\textcolor{blue}{庄闪闪的 R 语言手}

我是庄闪闪呀！欢迎关注我的公众号：庄闪闪的 R 语言手册。

我是庄闪闪呀！欢迎关注我的[公众号](#)：庄闪闪的 R 语言手册。

在上面的示例中，第一组花括号包含所需的文本颜色，第二组花括号包含应将此颜色应用到的文本。

#### 9.4.2 更改全文页边距等

在 top-level 中加入 `geometry` 命令，例如

```
---
```

```
title: "RMarkdown 常用技巧"
author:
  - 庄闪闪
documentclass: ctexart
geometry: "left=2cm,right=2cm,top=2cm,bottom=2cm"
output:
  rtticles::ctex:
    keep_tex: true
    includes:
      in_header: columns.tex
    fig_caption: yes
    number_sections: yes
    toc: yes
---
```

这时的页边距就变成下面这样了：

# RMarkdown 常用技巧

庄闪闪

## 目录

|               |   |
|---------------|---|
| 1 引言          | 1 |
| 引言            | 1 |
| 2 自我介绍        | 1 |
| 3 R 代码段       | 2 |
| 4 源代码控        | 3 |
| 5 小结          | 4 |
| 6 Two columns | 4 |

当然全文字体大小等操作也是这样操作的，在 geometry 操作即可：

TABLE 3.4: Available top-level YAML metadata variables for LaTeX output.

| Variable                                  | Description  |
|---|--|
| lang                                      | Document language code                                 |
| fontsize                                  | Font size (e.g., 10pt , 11pt , or 12pt )               |
| documentclass                             | LaTeX document class (e.g., article )                  |
| classoption                               | Options for documentclass (e.g., oneside )             |
| geometry                                  | Options for geometry class (e.g., margin=1in )         |
| mainfont, sansfont,<br>monofont, mathfont | Document fonts (works only with xelatex and lualatex ) |
| linkcolor, urlcolor,<br>citecolor         | Color for internal, external, and citation links       |

### 9.4.3 缩进文本

默认情况下，Markdown 还将忽略用于缩进的空格。但是在某些情况下，例如在经文和地址中，我们可能希望保留缩进。在这些情况下，我们可以通过以竖线（|）开头的线来使用线块。换行符和所有前导空格将保留在输出中。例如：

```
| When dollars appear it's a sign
|   that your code does not quite align
| Ensure that your math
|   in xaringan hath
|   been placed on a single long line
```

输出为：

When dollars appear it's a sign  
that your code does not quite align  
Ensure that your math  
in xaringan hath  
been placed on a single long line

#### 9.4.4 分页

如果想要分页，可以使用 `\newpage`。例如：如果想把目录和正文内容分开，可以在在正文前面加入这个代码

```
15 toc: yes
16 ---
17
18
19 \newpage
20
21 # 引言
22
23 中文LaTeX文档并非难题。当然这句话得站在巨人 [CTeX](http://ctex.org)
    的肩膀上才能说。它让我们只重写一句
```

这时输出的结果，目录一个界面，正文另起一页。

# 在 R Markdown 文档中使用中文

谢益辉 邱怡轩 于森

## 目录

|         |   |
|---------|---|
| 1 引言    | 2 |
| 2 字体和选项 | 2 |
| 3 R 代码段 | 3 |
| 4 源代码控  | 4 |
| 5 小结    | 4 |

### 9.4.5 控制文本输出的宽度

有时从 R 代码输出的文本可能太宽。如果输出文档具有固定的页面宽度（例如，PDF 文档），则文本输出可能会超过页面的页边距。

R 全局选项宽度可用于控制 R 函数输出的文本宽度，如果默认值太大，则可以尝试使用较小的值。此选项通常表示每行字符的粗略数目。例如：

```
```{r}
options(width = 300)
matrix(runif(100), ncol = 20)
```

```
```
```

```
options(width = 300)
matrix(runif(100), ncol = 20)

##          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]      [,10]      [,11]      [,12]      [,13]      [,14]      [,15]
## [1,] 0.6090561 0.7827552 0.1173482 0.5025686 0.8344960 0.9784600 0.004335023 0.99936
## [2,] 0.1655875 0.8415561 0.5318202 0.3361056 0.1679164 0.4238841 0.315499210 0.72263
## [3,] 0.7391834 0.9233610 0.8280438 0.4163201 0.6271172 0.7652541 0.001322856 0.67880
## [4,] 0.2664443 0.6543799 0.8576013 0.9486274 0.5458728 0.7163018 0.887708630 0.29174
## [5,] 0.6480914 0.3429590 0.4333605 0.2704876 0.3063819 0.2630814 0.536264875 0.57021
```

```
```{r}
```

```
options(width = 60)
matrix(runif(100), ncol = 20)
```

```

```
options(width = 60)
matrix(runif(100), ncol = 20)

##          [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.6227367 0.2070252 0.68505433 0.3527539 0.34823954
## [2,] 0.8047349 0.3282919 0.69252156 0.5567780 0.29265065
## [3,] 0.7610162 0.4288263 0.04660892 0.7272211 0.07826963
## [4,] 0.5212527 0.9683995 0.39185830 0.5156720 0.83988932
## [5,] 0.3696163 0.1574025 0.23380071 0.9789555 0.68128499
##          [,6]      [,7]      [,8]      [,9]      [,10]
## [1,] 0.8552318 0.01785735 0.38319809 0.10746410 0.7790023
## [2,] 0.5652474 0.96712150 0.07648612 0.07842289 0.5551644
## [3,] 0.5968318 0.27098671 0.36053489 0.12384635 0.8608930
## [4,] 0.1218191 0.11291128 0.15235481 0.85970190 0.6659506
## [5,] 0.7888682 0.27392856 0.23954630 0.29620253 0.6205482
##          [,11]      [,12]      [,13]      [,14]      [,15]
```

但是这种方式不一定对所有函数都适用，这是你可以使用其他方式，对于 Html（这里不做解释，主要将 pdf），可以参见教程。

对于 PDF 输出，换行比较困难。一种解决方案是使用 LaTeX 软件包清单，可以通过 Pandoc 参数 `-listings` 启用它。然后，您必须为此软件包设置一个选项，并且可以从外部 LaTeX 文件中包含设置代码（有关详细信息，请参见第6.1节），例如，

```
---
```

```
output:
```

```
pdf_document:  
pandoc_args: --listings  
includes:  
in_header: preamble.tex  
---
```

在 preamble.tex 中（建议放到和 rmd 同一个文件夹），我们设置了 Listings 包的一个选项：

```
\lstset{  
breaklines=true  
}
```

这是输出的结果，但是其实不是很美观

```
options(width = 300)
matrix(runif(100), ncol = 20)

##          [,1]      [,2]      [,3]      [,4]
## [1,] 0.06587866 0.40192875 0.5999033 0.7315387
## [2,] 0.5623659 0.6306781 0.1082544 0.2070100 0.5135456
## [3,] 0.6416327 0.51111402 0.74581276 0.1458475 0.1626146
## [4,] 0.42534567 0.05344768 0.06217746 0.6549950
## [5,] 0.07300677 0.22077002
## [6,] 0.38971541 0.88928213 0.9976609 0.7738947
## [7,] 0.0780520 0.5484644 0.7777476 0.9676629 0.5223810
## [8,] 0.6300901 0.08484612 0.91139206 0.5441875 0.1178050
## [9,] 0.74314202 0.12070480 0.47943594 0.2622410
## [10,] 0.71769999 0.13293773
## [11,] 0.46872228 0.98226813 0.5368637 0.4250215
## [12,] 0.4307330 0.4797191 0.2177014 0.5037882 0.6394169
## [13,] 0.5560922 0.08762501 0.93214973 0.1293005 0.5585630
## [14,] 0.44646346 0.69875931 0.05881515 0.9334706
## [15,] 0.20124937 0.78773379
## [16,] 0.65166468 0.07313946 0.8366737 0.9932661
## [17,] 0.1208101 0.9141263 0.5431332 0.4456278 0.7204060
## [18,] 0.4240268 0.79148886 0.03824945 0.7535966 0.8420234
## [19,] 0.44190815 0.92664651 0.70767045 0.9472987
```

#### 9.4.6 控制图片输出大小

这个我在文稿和教程中说的挺清楚了。一共有两种方式：

方法一：

```
```{r, echo=FALSE, out.width="50%", fig.cap="A nice image."}
knitr::include_graphics("foo/bar.png")
```

```

方法二：

```
![A nice image.](foo/bar.png){width=50%}
```

### 9.4.7 图片对齐

这个我也说过啦！可见：R 沟通 | Rmarkdown 教程（3），R 沟通 | Rmarkdown 教程（2）。如果使用 R 代码导入图片的话，使用 `knitr::include_graphics()` 并结合 R chunk 中 `fig.align = 'center'` 参数进行居中。如果结果不想显示代码块，可在 chunck 中加入：`echo=FALSE`。

任何输出形式都适用，推荐使用

```
knitr::include_graphics()
```

当然还有另一种方法，不使用代码形式。对于输出为 html，你可以使用 html 语法（不适用于 pdf/word）

```
<center>  </center>
```

对于输出 pdf/word 可以使用以下方式

```
\center

\center
```

当然想要文字居中的话，也是这样使用。### 5.8. 代码块的行号

通过块选项 `attr.source = ".numberLines"` 将行号添加到源代码块中，或者通过 `attr.output = ".numberLines"` 将文本输出块添加到文本中，例如，

```
```{r, attr.source=".numberLines"}
if (TRUE) {
  x = 1:10
  x + 1
}
```

```

输出结果为：

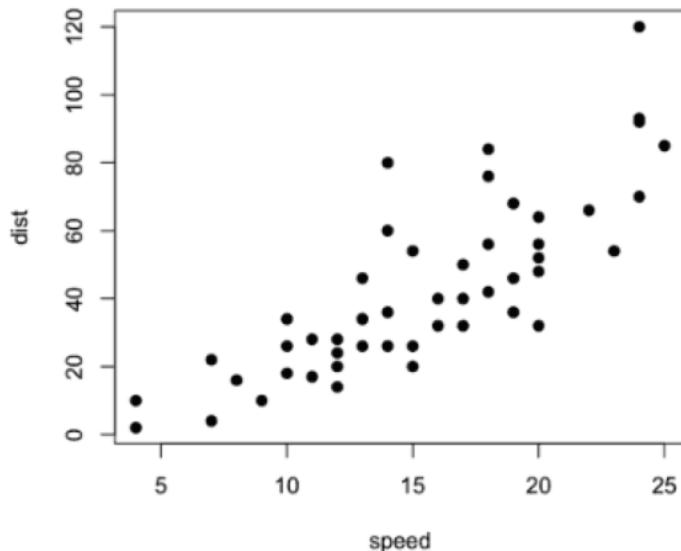
```
1 if (TRUE) {  
2   x <- 1:10  
3   x + 1  
4 }  
  
## [1] 2 3 4 5 6 7 8 9 10 11
```

#### 9.4.8 多列输出

这个特别好用！虽然学起来有那么一点困难，具体我再出一期推文，把这个讲清楚。具体可以见这里的教程。类似于排版成这种形式：

## Two columns

Below is a Div containing three child Divs side by side. The Div in the middle is empty, just to add more space between the left and right Divs.



The figure on the left-hand side shows the `cars` data.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.



# Chapter 10

## 其他方式的文档沟通

### 10.1 Typora

markdown 的轻量级编辑器，非常推荐大家用它做笔记。教程从上到下进行拓展，首先了解 Typora 的特性，然后对其中存在的问题进行解决。这里包括：设置免费图床，使得 Typora 中的图同步到网上；给 Typora 中的字体变化颜色。当然，以后也会继续拓展。

R 沟通 | markdown 编辑器—Typora

R 沟通 | 如何在 Typora 中设置免费的图床

R 沟通 | Typora 字体颜色设置

### 10.2 xaringan 幻灯片

强烈推荐使用这个包制作幻灯片，难度从上往下一步步晋升。认真学了这三期基本够用。小编带你装 x，做出不一样的幻灯片。建议大家先把前面的入门教程过一遍，再往后学。

R 沟通 | 用 xaringan 包制作幻灯片

R 沟通 | 设置 xaringan 主题

R 沟通 | 提升 xaringan 幻灯片的 b 格

### 10.3 制作幻灯片

如果想要小编制作的 ppt 模板可以后台私聊。比起传统的 ppt，最大的优点就是能够将代码和结果直接输出到 ppt 中。

R 沟通 | 制作个性化 ppt!

## 10.4 Rbeamer

学术报告可以使用它，类似于 LaTeX 中的 Beamer 模板。小编基本都是使用它做讨论班的幻灯片（数学公式，代码，结果相结合）。

R 沟通 | 舍弃 Latex，拥抱 Rbeamer 吧！

## 10.5 Dashboards

使用 markdown 语法构建可交互网站。非常容易入门，如果你对 shiny 一窍不通的话，建议先使用它来完成你的初步项目吧。

R 文档沟通 | Dashboards 入门 (1)

R 文档沟通 | Dashboards 入门 (2)

R 文档沟通 | Dashboards 入门 (3)

## 10.6 shiny

该系列转自于公众号《生信技能树》，论 shiny 教程和功力，他们更优秀，我就不再出教程啦。我也是个小白，学习者。

基于 R 语言的 shiny 网页工具开发基础系列-01

基于 R 语言的 shiny 网页工具开发基础系列-02

基于 R 语言的 shiny 网页工具开发基础系列-03

基于 R 语言的 shiny 网页工具开发基础系列-04

基于 R 语言的 shiny 网页工具开发基础系列-05

基于 R 语言的 shiny 网页工具开发基础系列-06

基于 R 语言的 shiny 网页工具开发基础系列-07

## 10.7 blogdown

使用 blogdown 搭建私人博客。基础版本推文已经在《统计之都》发送了，可见：手把手带你搭建个人博客。

## 10.8 制作简历

R 沟通 | 使用 latex 模板构建个人履历

## 10.9 bookdown

R 沟通 | Bookdown 中文书稿写作手册

R 沟通 | 用 bookdown 制作图书 (1)

## 10.10 小编有话说

1. 最近有空会把写过推文的教程录制成视频，并加入 b 站**Rmarkdown** 从入门到精通教程中。
2. 还会加入其他有关文档沟通的推文，**bookdown**, **blogdown**, **pagedown** 等，以及对 Rmarkdown 做更加进一步的拓展。
3. 当然欢迎更多的读者提出宝贵的意见，或者有好的想法也欢迎和小编沟通交流。



# Chapter 11

## 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，

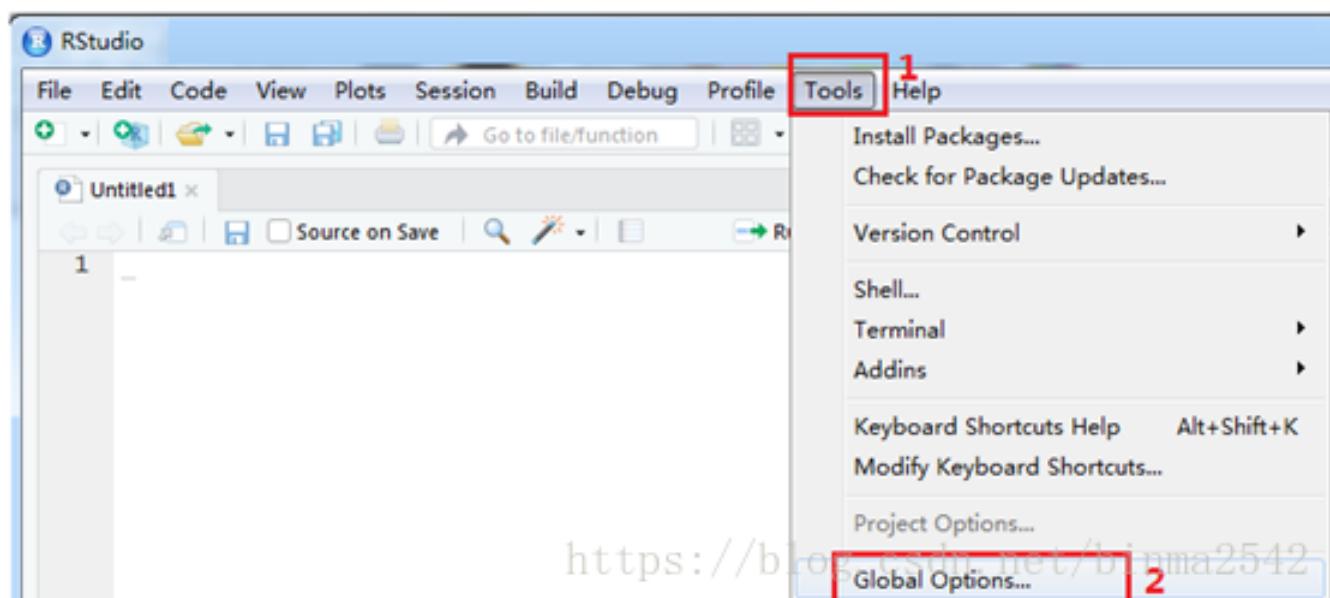


图 11.1: 选择国内镜像 1

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'
```

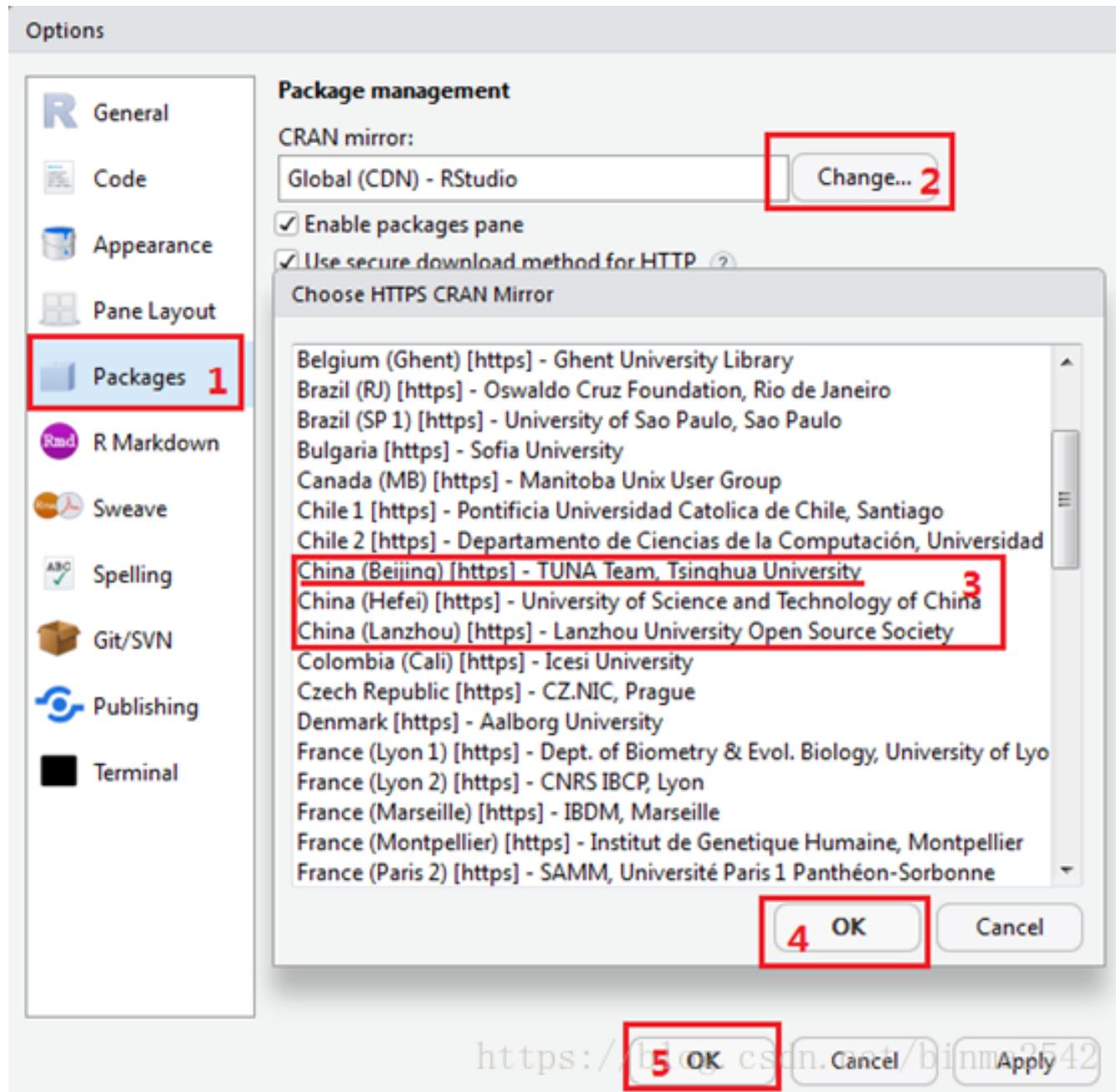


图 11.2: 选择国内镜像 2

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

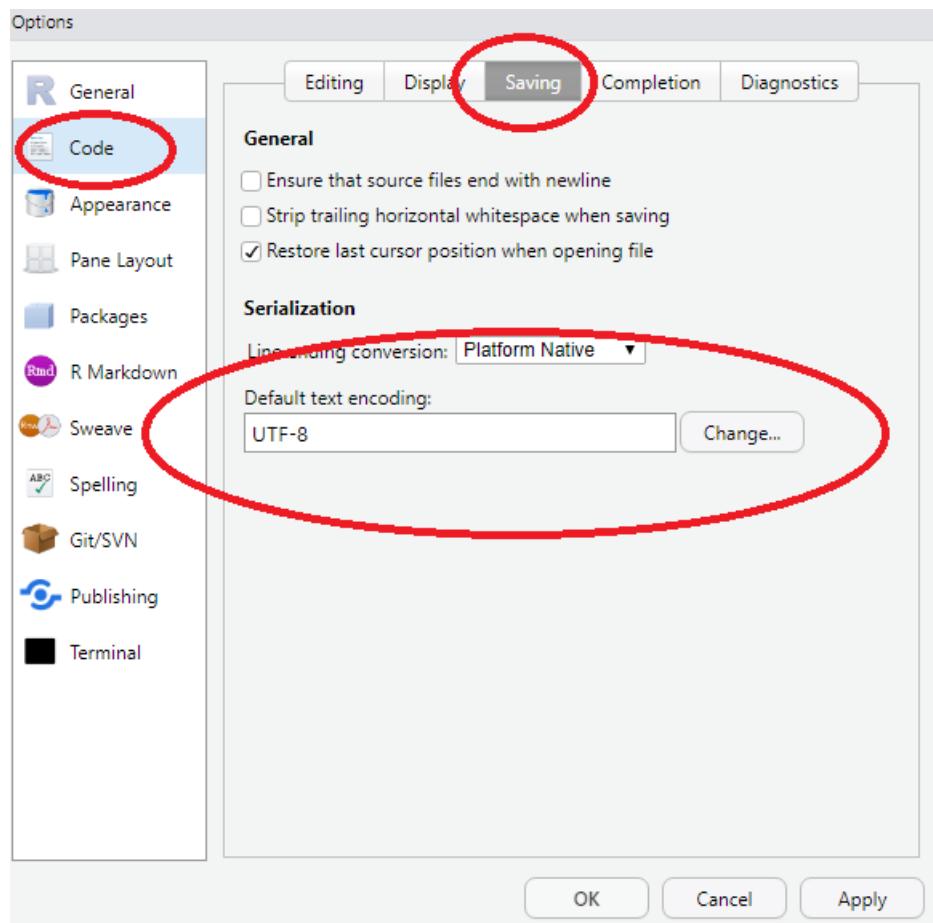


图 11.3: 代码是乱码

- 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

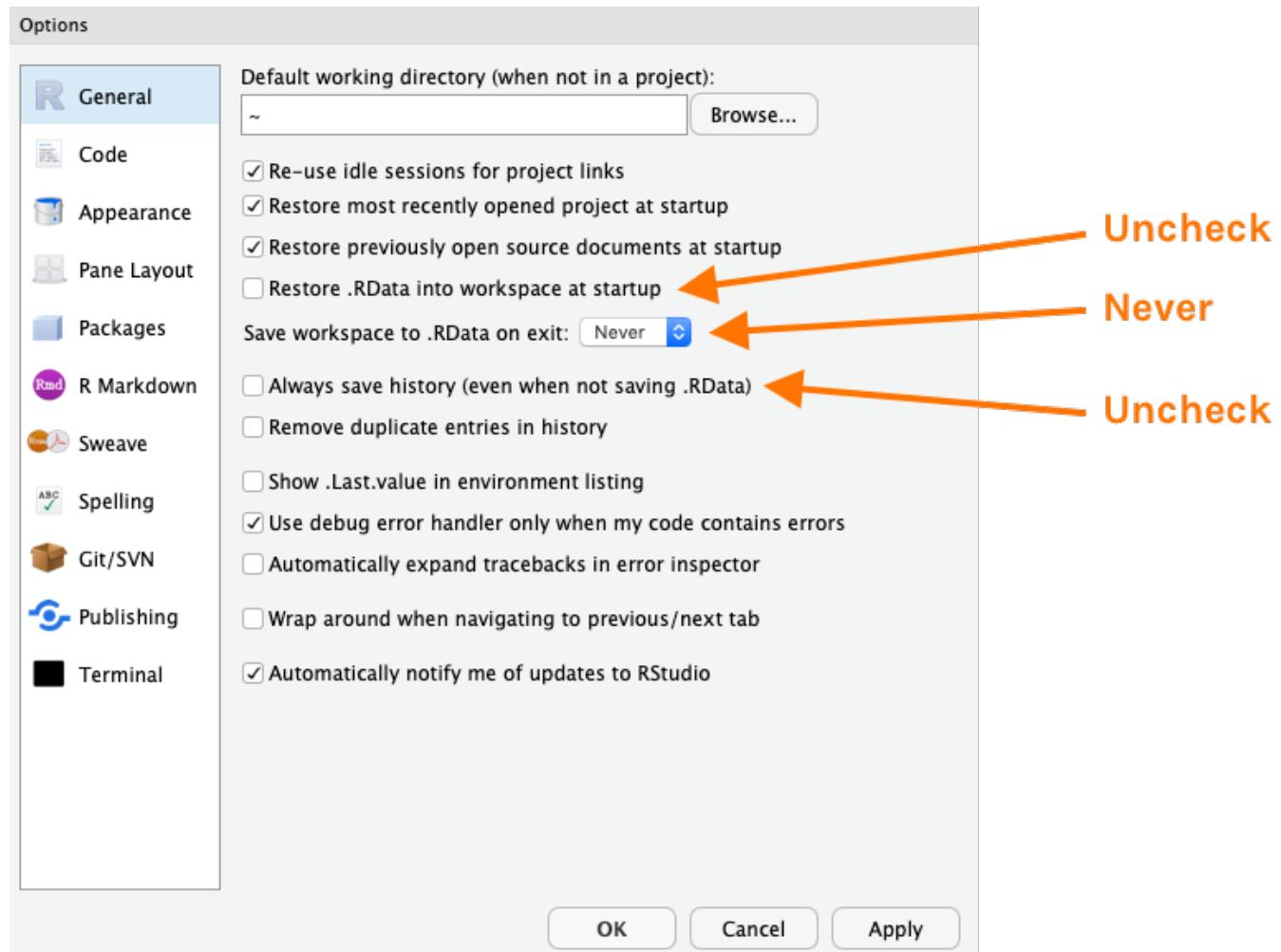


图 11.4: 打开 Rstudio 非常慢

- 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](#)。

## 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：

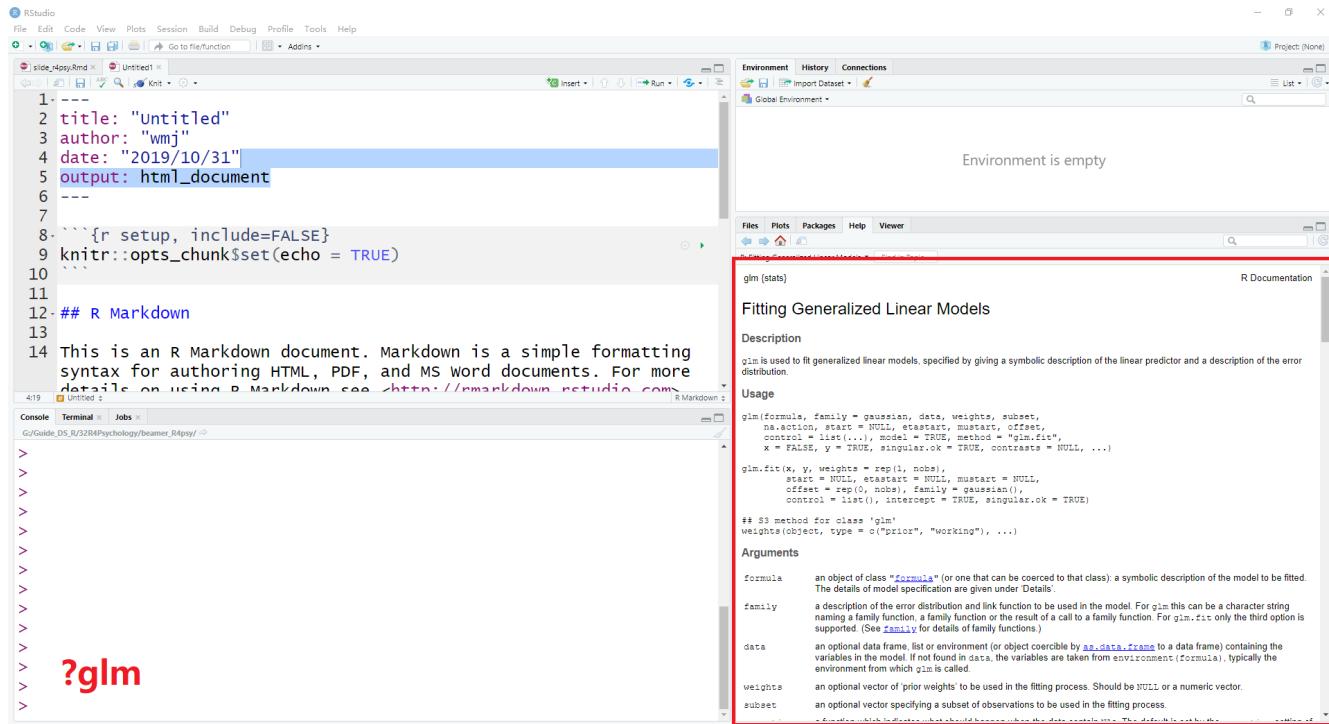


图 11.5: help 帮助

## 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 12

## 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 12.1: 个人公众号: 庄闪闪的 R 语言手册

下午3:59

**庄闪闪的R语言手册 ★**

在读应用统计研究生，R语言爱好者。来跟着我一起学R数据科学，可视化。

71篇原创内容 225位朋友关注

进入公众号 不再关注

可视化 数据科学

昨天晚上11:48

R语言学习

R语言数据可视化

2020总结, 2021展望

公众号推文分类目录

数据科学 R包 统计理论 分享资料

ggridges包—峰峦图详细介绍

原创:庄闪闪

esquisse包—不写代码生成ggplot图

原创:庄闪闪

calendR包—私人定制专属日历

原创:庄闪闪

corrplot包：相关性矩阵可视化

原创:庄闪闪

可视化 其他

R可视乎 | 三元相图

R可视乎|2021年日历大派送

R可视乎|瀑布图

R可视乎|等高线图

R可视乎|等高线图

31位朋友看过

Rstudio 界面介绍 (上)

Rstudio 界面详细介绍，各个窗口详细介绍，包括：下载安装更新包；导入各类数据；绘图窗口介绍；历史代码查询等。

小编第一次录视频，多多见谅，以后继续改进。

周一 20:00

其他联系方式 (可跳转):

- 知乎
- Github
- CSDN
- b 站