R 语言初级绘图

（公众号：庄闪闪的 R 语言手册）

庄闪闪

2021-01-26

目录

1 绘制基本图形

1.1 绘制分布关系

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

主要方法：绘制相应的图形，如直方图，条形图、饼图、箱线图等。

1.1.1 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。

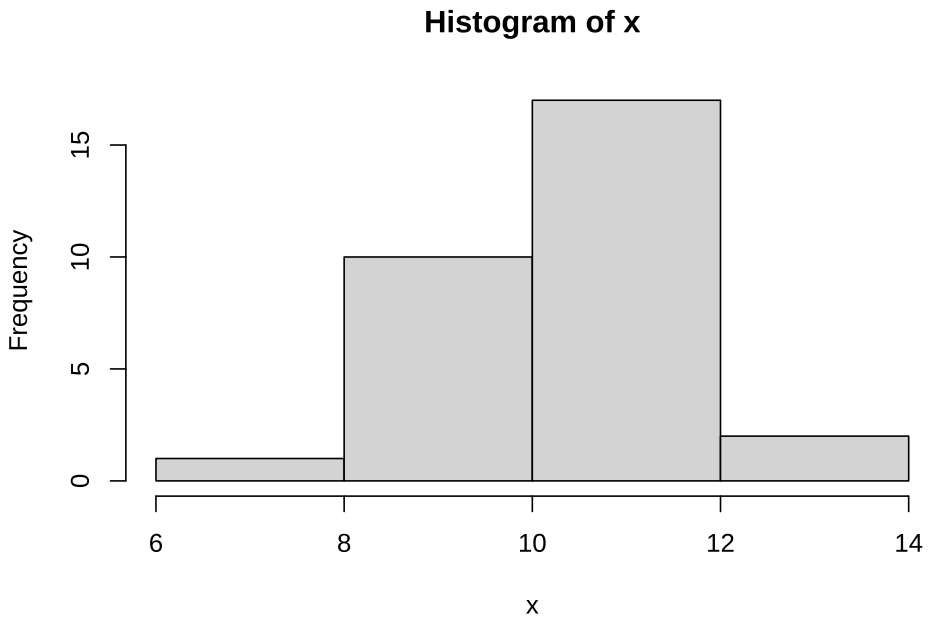
例子：我们使用模拟数据进行讲解，通过正态分布产生 30 个数据。

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

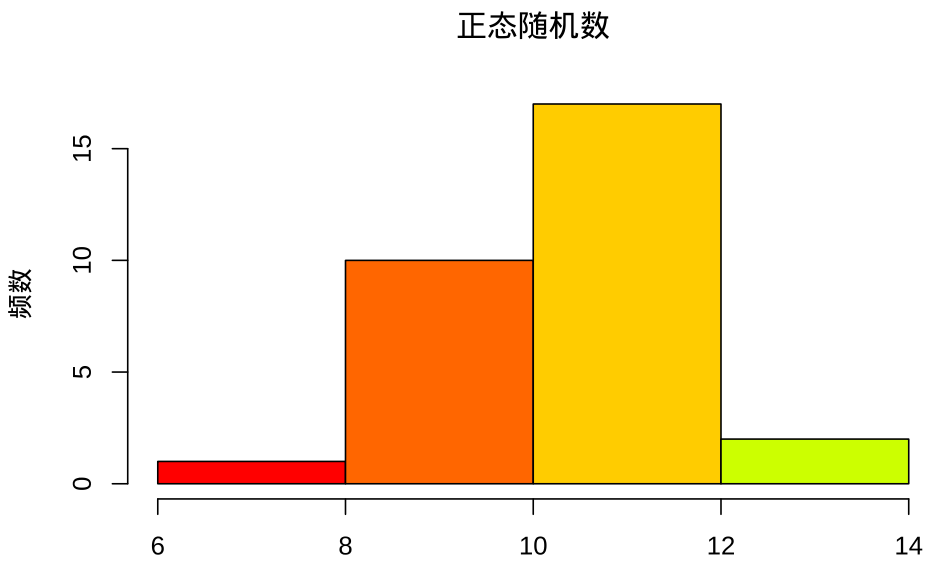
```
##  [1] 10.60 10.61 10.08  9.56  9.29 10.94 10.64 10.41  7.98 11.11 12.08 12.56
## [13] 11.46 11.06  9.28 10.31  9.48 10.12  8.23  9.65  9.28  9.75  9.81 10.01
## [25] 10.84  9.21 10.72 11.44 10.98 10.11
```

`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')
```

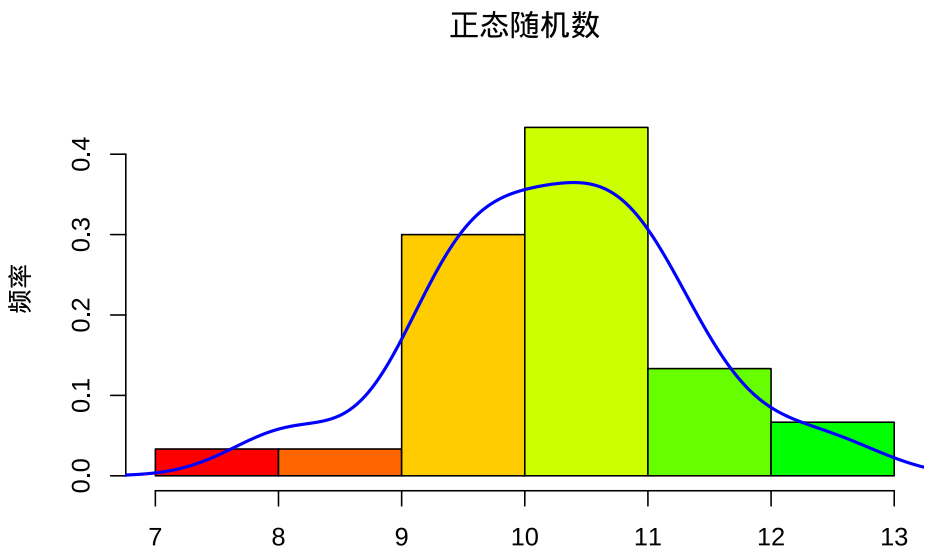


`breaks = 3` 表示 x 轴分为 3 个节点。其他设置可参考帮助文档，即`?hist`。这里加入其他参数 `col`，`main`，`xlab`，`ylab`，分别表示颜色，主题名称，x

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')
```

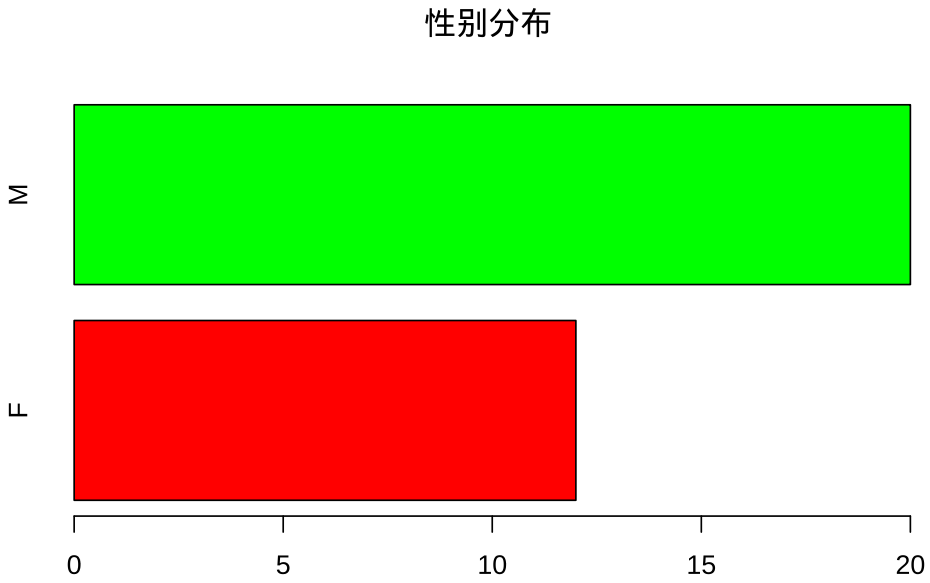


1.1.2 条形图

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。

```r
# 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)
```



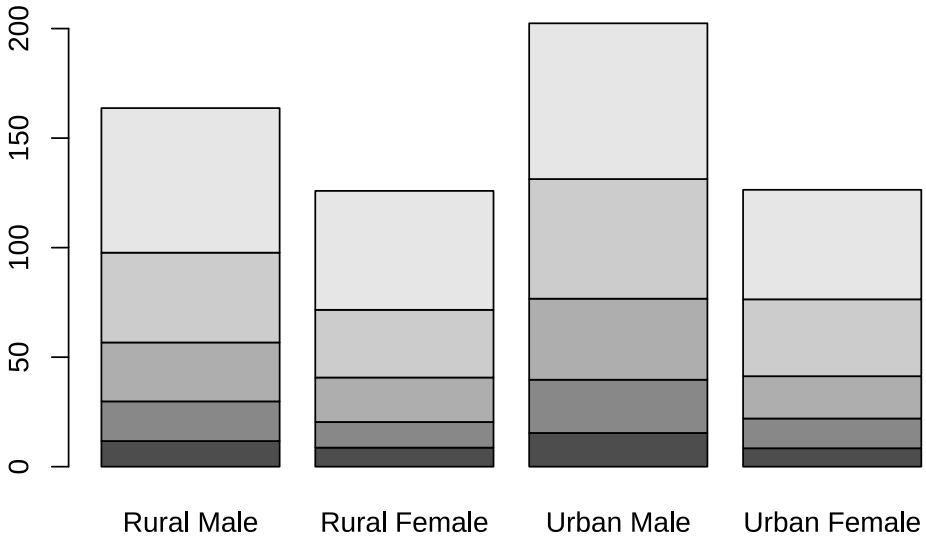
数据介绍：VADeaths 数据集记录的是 1940 年 Viginia(弗吉尼亚洲) 不同人群（Rural Male、Rural Female 、Urban Male、Urban Female）中每一千人的死亡情况。

例子：数据前 6 行展示如下：

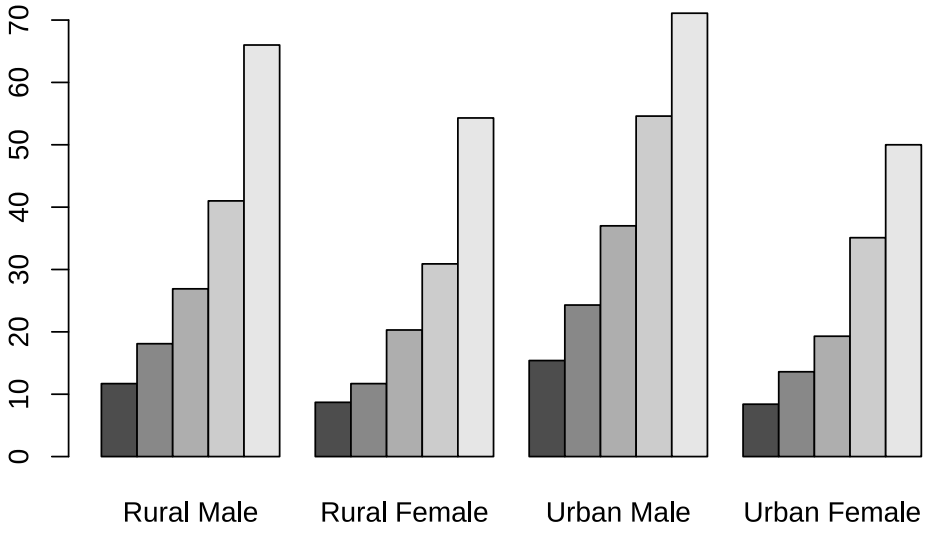
	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

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)
```



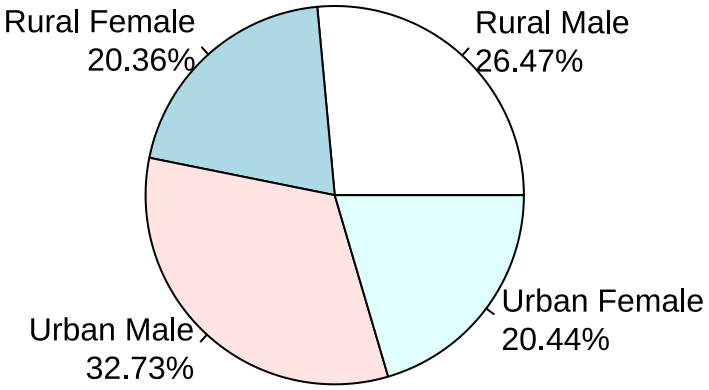
结论：随着年龄的增长，`Viginia` 人群的死亡率逐渐增加，并且在 4 类人群中，`Urban Male` 的死亡率比同年龄段的其他群体的死亡率高。同时，在同一环境下，相同年龄段的男性的死亡率要比女性高。

1.1.3 饼图

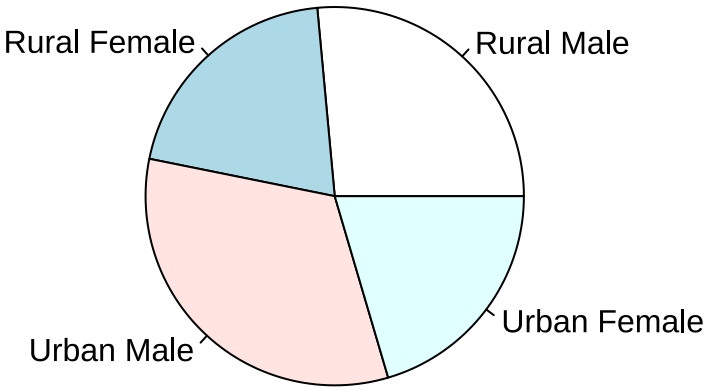
概念介绍：将各项的大小与各项总和的比例。反映部分与部分、部分与整体之间的比例关系。

例子：

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




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




```
# ?pie
```

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

1.1.4 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。

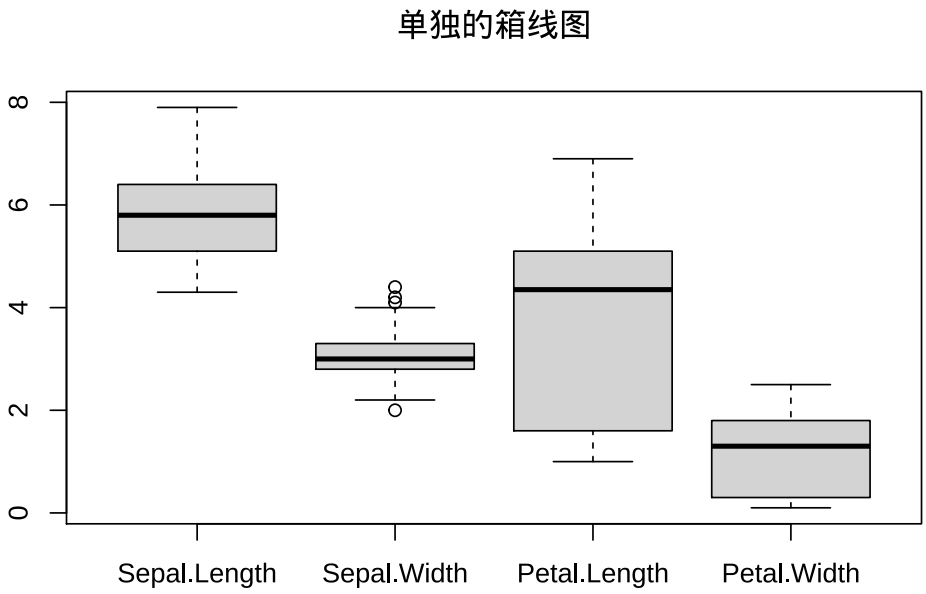
数据介绍：iris 数据集（鸢尾花数据集），是常用的分类实验数据集合，由 Fisher 在 1936 年收集整理。数据集包含了 150 个子数据集，分为 3 类（分别为 setosa、versicolor、virginica），每类 50 个数据，每个数据包含 4 个属性，即花萼长度 Sepal.Length 、花萼宽度 Sepal.Width、花瓣长度 Petal.Length、花瓣宽度 Petal.Width。

前 6 行数据如下：

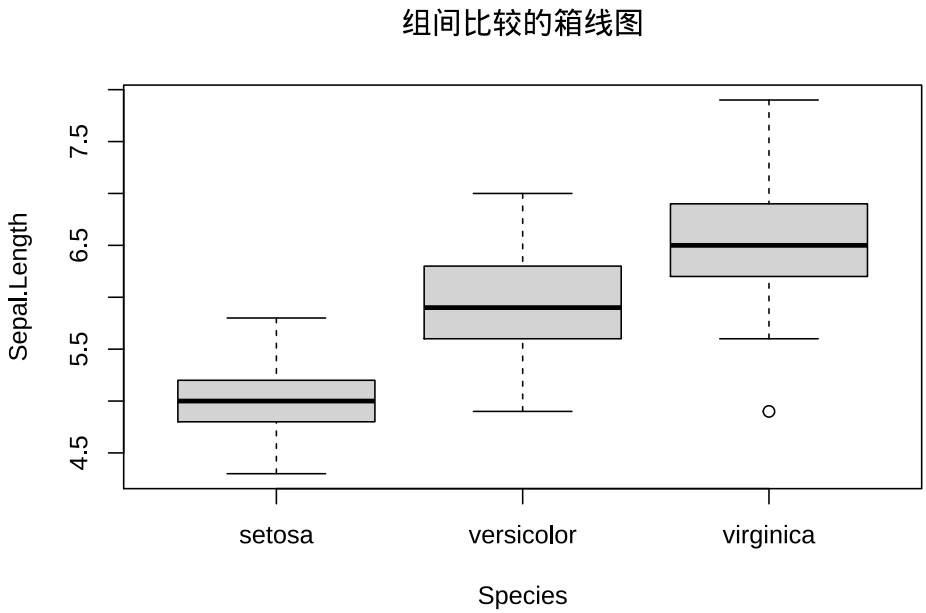
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

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')
```




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



结论：第一个图：Sepal.Width 列含有四个异常值。

第二个图：Sepal.Length 列中，类别属于 virginica 的数据含有一个异常值。

同时，从第一个图可以看到，Petal.Length 列前半部分相对分散，而后半部分相对密集。

1.2 绘制数据间关系

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

1.2.1 散点图

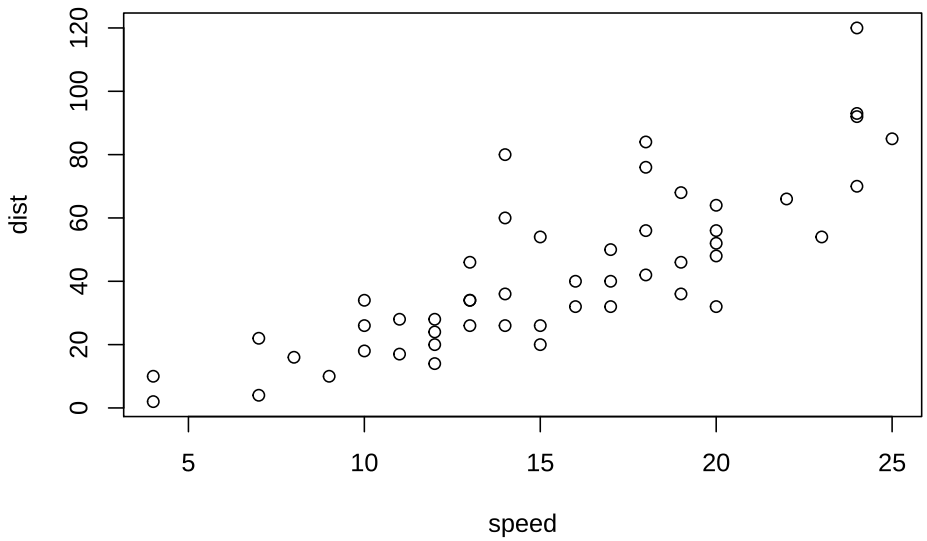
特点：

(1) 特征之间是否存在关联趋势，关联趋势是线性的还是非线性的。

(2) 一目了然的看出离群值。从而可以进一步分析这些离群值是否可能在建模分析中产生很大的影响。

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

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



```
# plot(cars)  # 效果同上
```

结论：随着汽车行驶速度的增加，刹车距离也在不断增加。

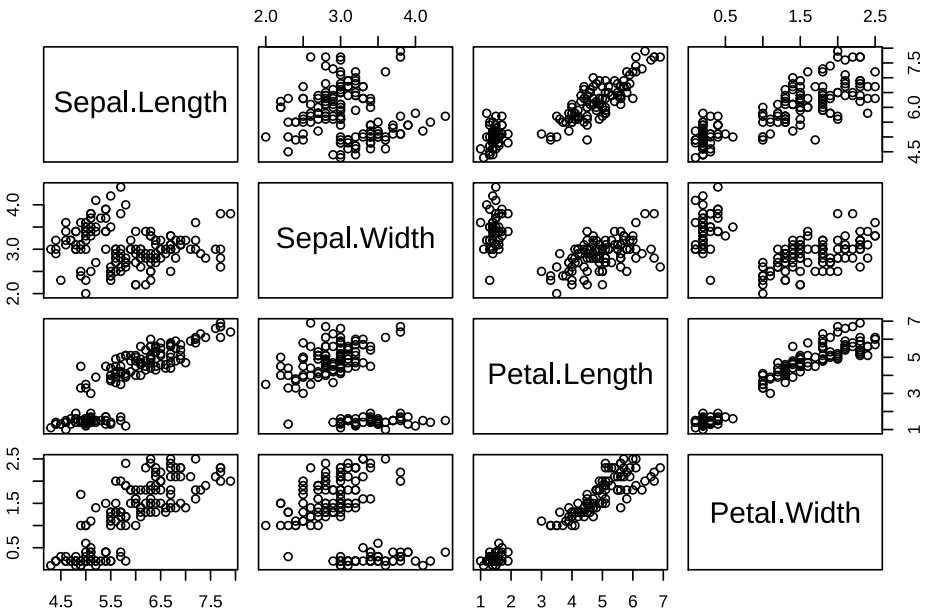
1.2.2 散点矩阵图

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

适用：高维数据

- **plot**

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

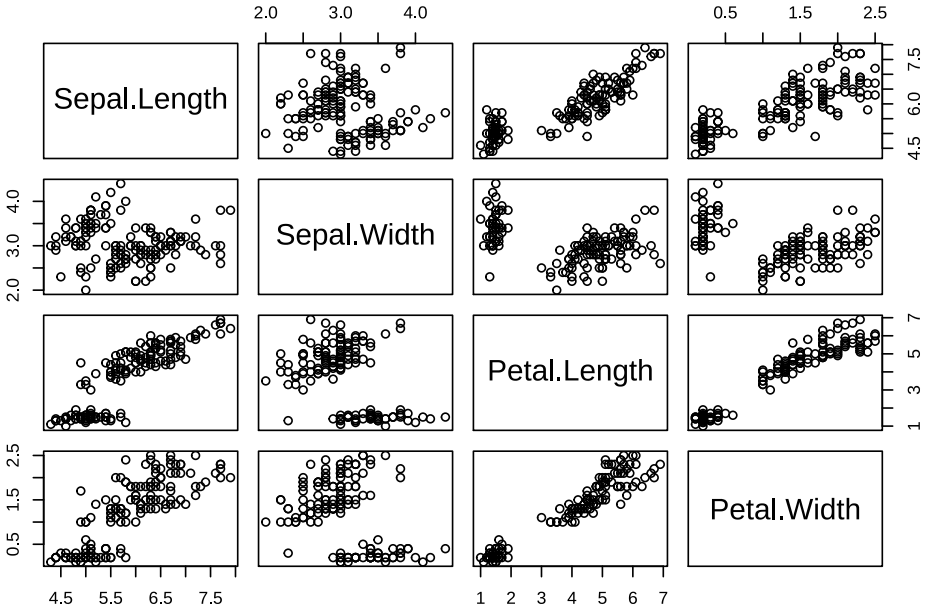


结论：花瓣长度（Petal.length）与花瓣宽度（Petal.Width）有明显的线性关系，其余属性之间的关系不是很明显。

- **pairs**

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

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



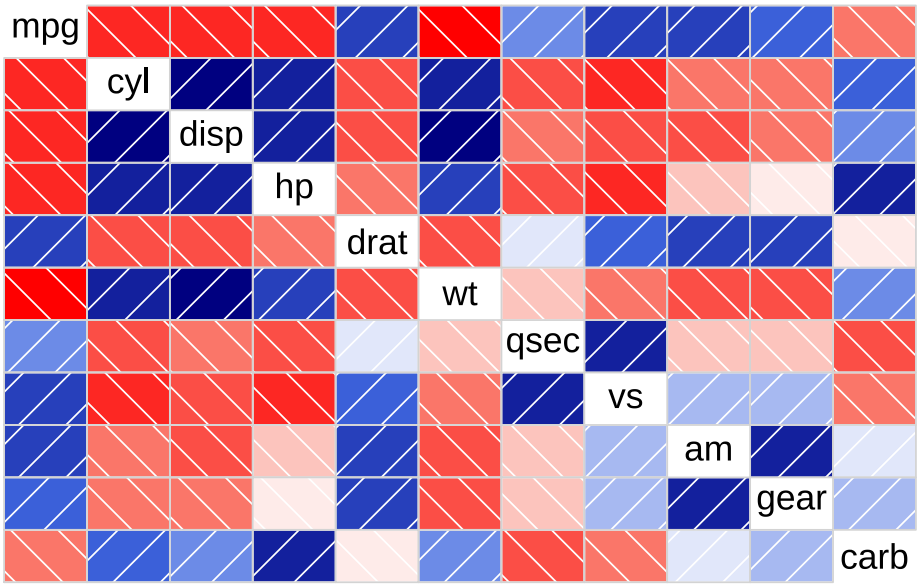
1.2.3 多变量相关矩阵图

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相关关系，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。

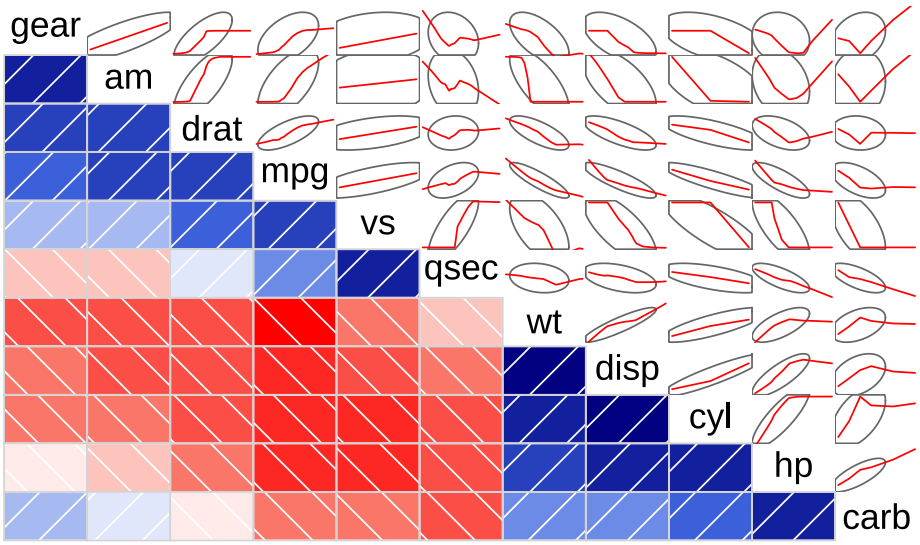
数据介绍：Mtcar 数据集是 1974 年 Motor Trend US 杂志公布的 32 辆车的 11 个数据，包括燃料消耗和 10 个关于汽车设计与性能的数据。

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")
```

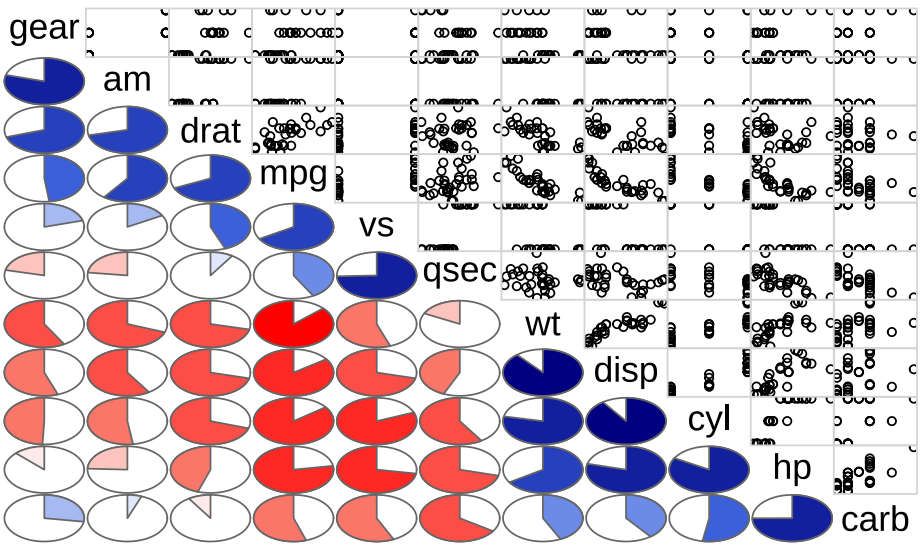



```
# 相关图，主对角线上方绘制置信椭圆和平滑拟合曲线，主对角线下方绘制阴影
```

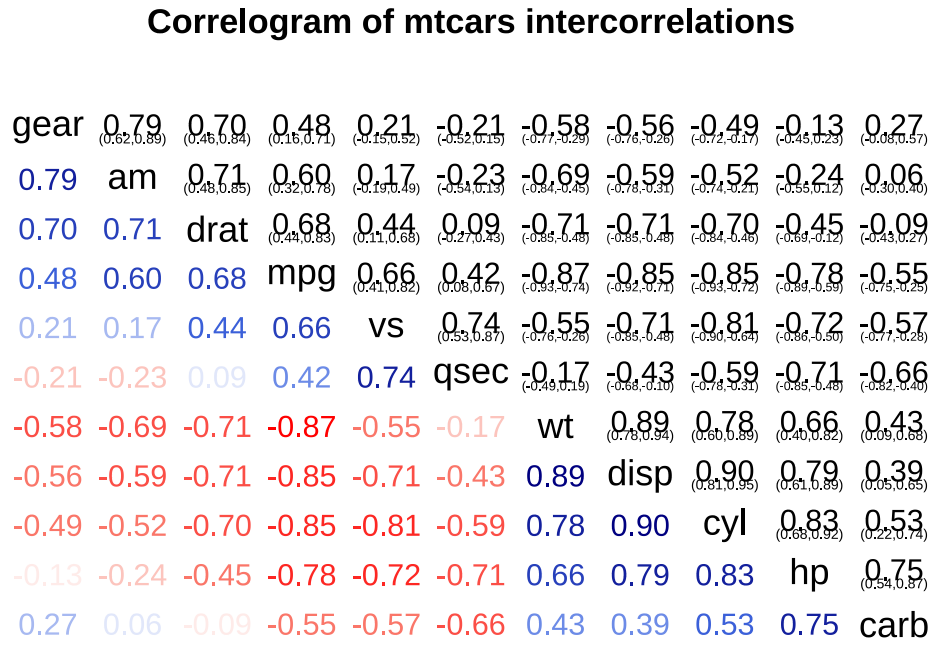
结论：disp 与 cyl 呈正相关关系，且相关程度较高。此外，mpg 与 wt 呈高度负相关，且 am 与 carb 基本没有关系。

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")
```



1.3 绘制其他图形

1.3.1 核密度图

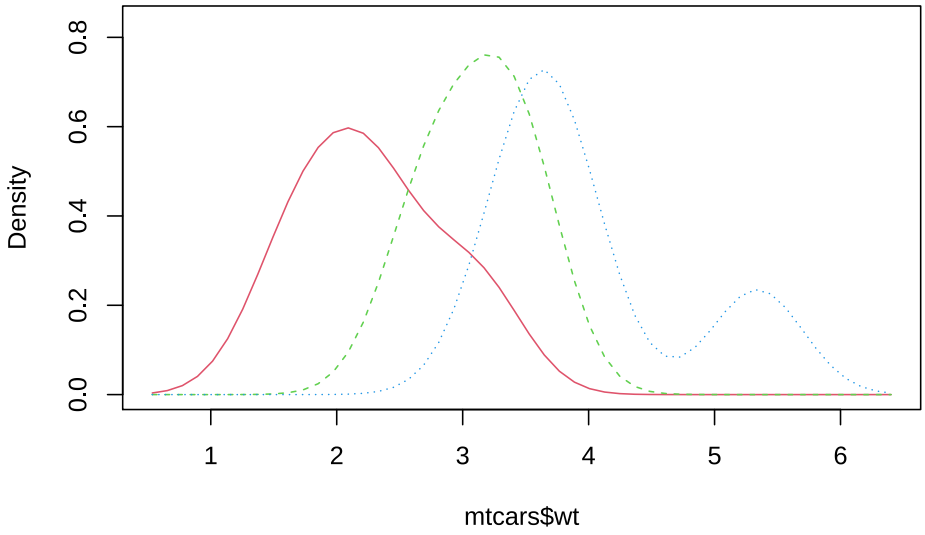
概念介绍：sm 包中 `sm.density.compare` 函数用于绘制核密度图，核密度图如果想用一条密度曲线而不是通过柱状来展示连续型变量的分布。

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

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

其中 x 是数值向量，group 是分组向量，是因子型数据。

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



1.3.2 小提琴图

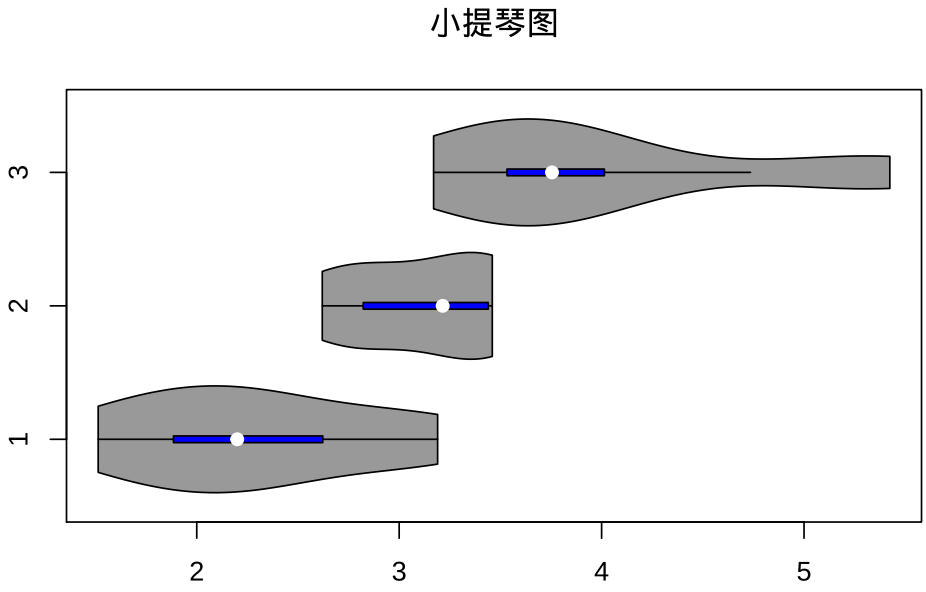
概念介绍：vioplot 包中的 vioplot 函数用于绘制小提琴图，小提琴图是核密度图与箱线图的结合，本质是利用密度值生成的多边形，但该多边形同时还沿着一条直线作了另一半对称的“镜像”，这样两个左右或上下对称的多边形拼起来就形成了小提琴图的主体部分，最后一个箱线图也会被添加在小提琴的中轴线上。

使用格式如下。

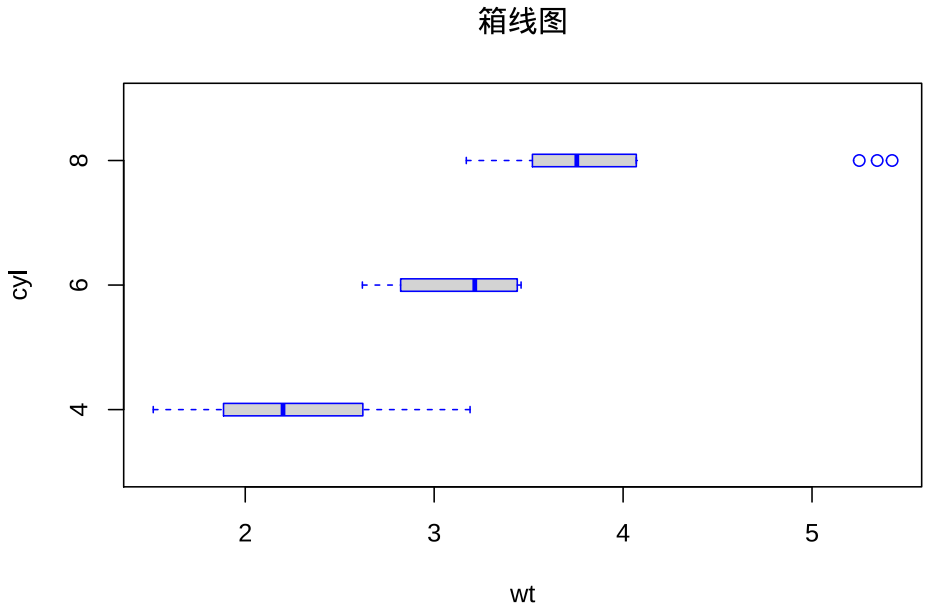
```
vioplot( x , ..., range=1.5, h, ylim, names, horizontal=FALSE
, …)
```

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
```




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



1.3.3 QQ 图

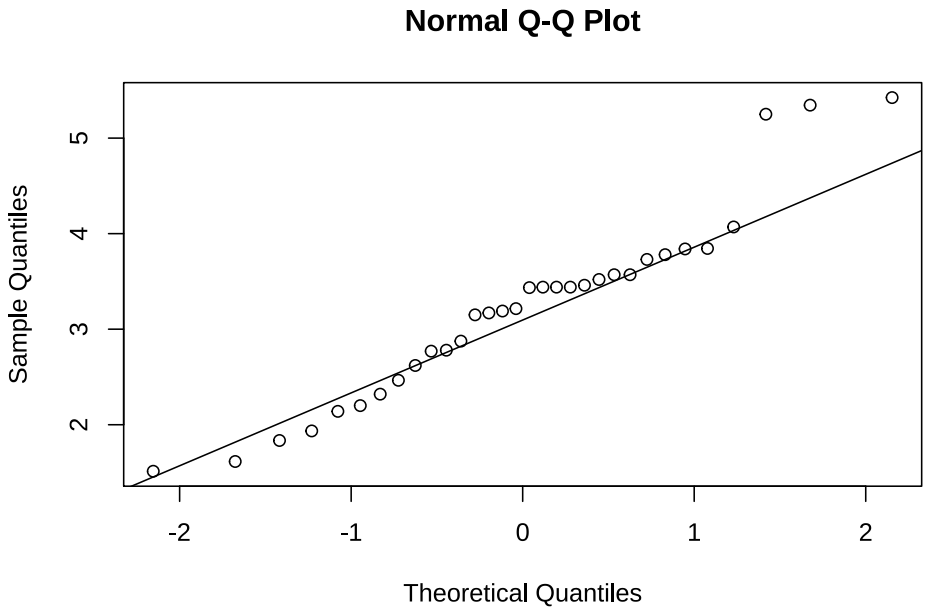
概念介绍：查看经验分布和理论分布是否一致。将排序后的数据和理论分布的分位数进行比较后大致相等，说明了经验分布和理论分布相似。`qqplot()` 函数用于绘制 QQ 图，QQ 图检查数据是否服从某种分布。

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
```



1.3.4 等高图

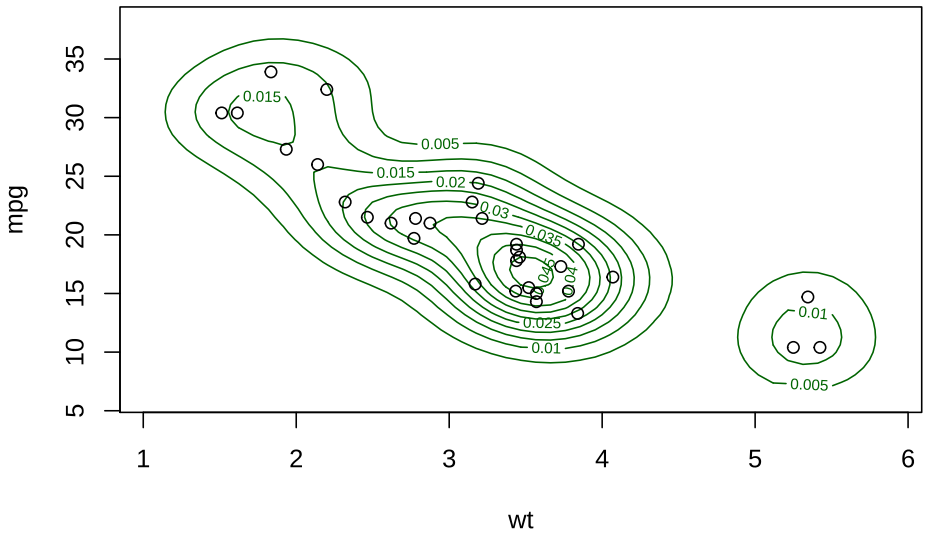
概念介绍：数据形式：两个数值向量 x、y 和一个相应的矩阵 z。x、y 交叉组合之后形成的是一个“网格”，z 是这个网格上的高度数值，将平面上对应的 z 值（高度）相等的点连接起来形成的线就是等高线。对 x、y 进行核

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
```



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。

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:

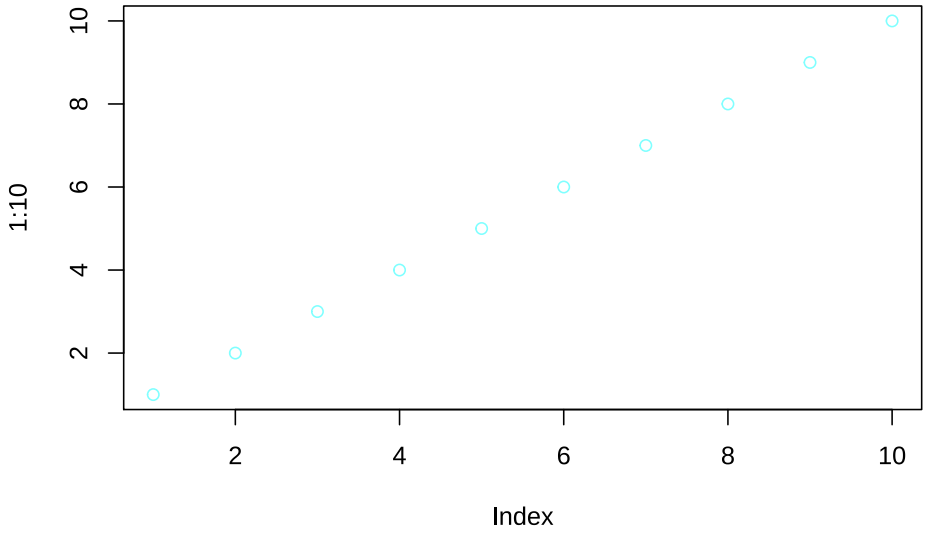
2.1.1 固定颜色选择函数

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))
```

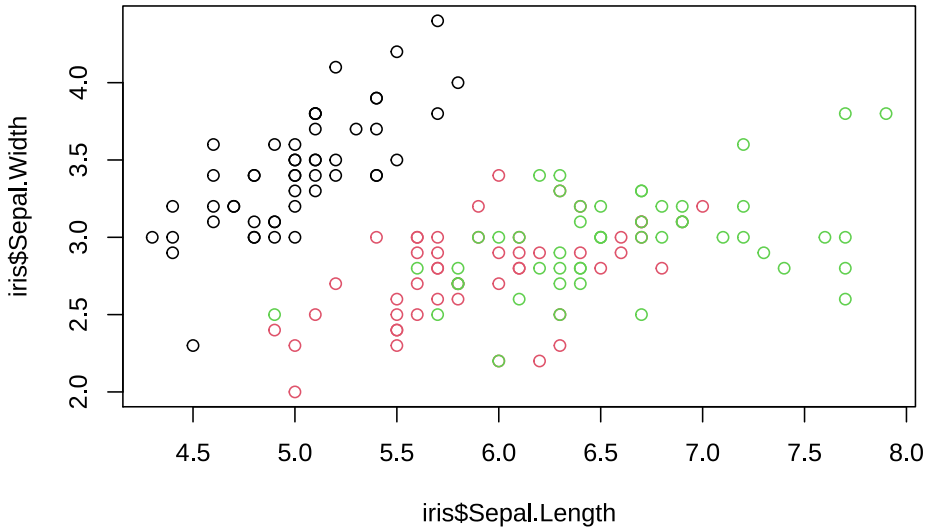


通过 palette 函数固定调色板，只要设定好了调色板，它的取值就不会再改变（直到下一次重新设定调色板）。

```
palette() # 返回当前的调色板设置，此时为默认值
## [1] "black"   "#DF536B" "#61D04F" "#2297E6" "#28E2E5" "#CD0BBC" "#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)
```

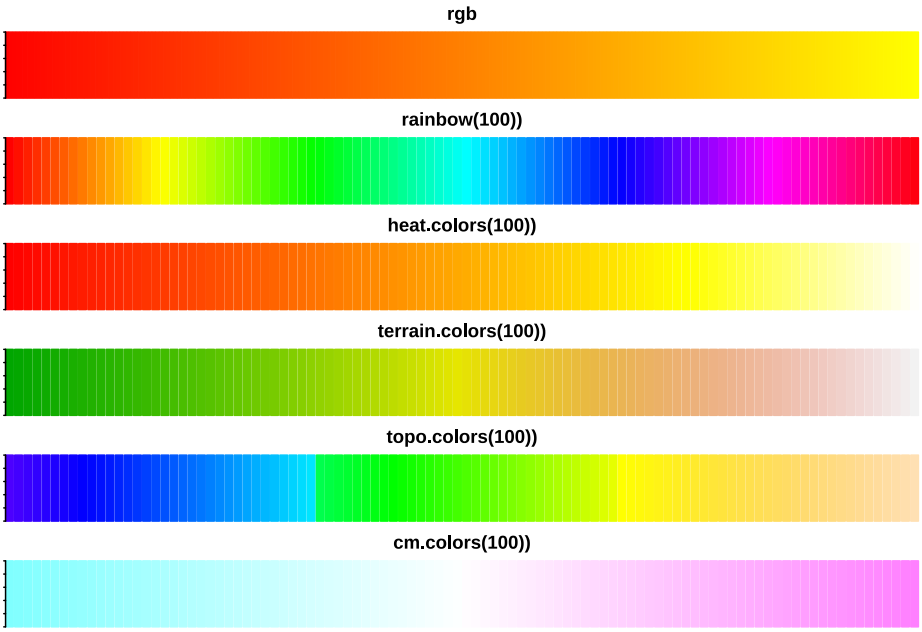
2.1.2 渐变色生成函数

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

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

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

`rainbow` 函数、`heat.coclor` 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。

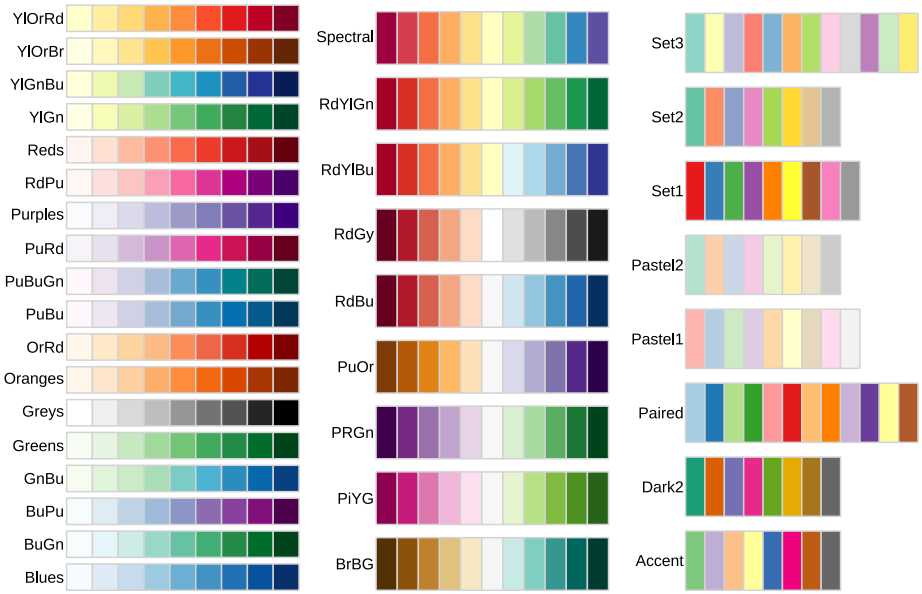


2.1.3 RColorBrewer 包

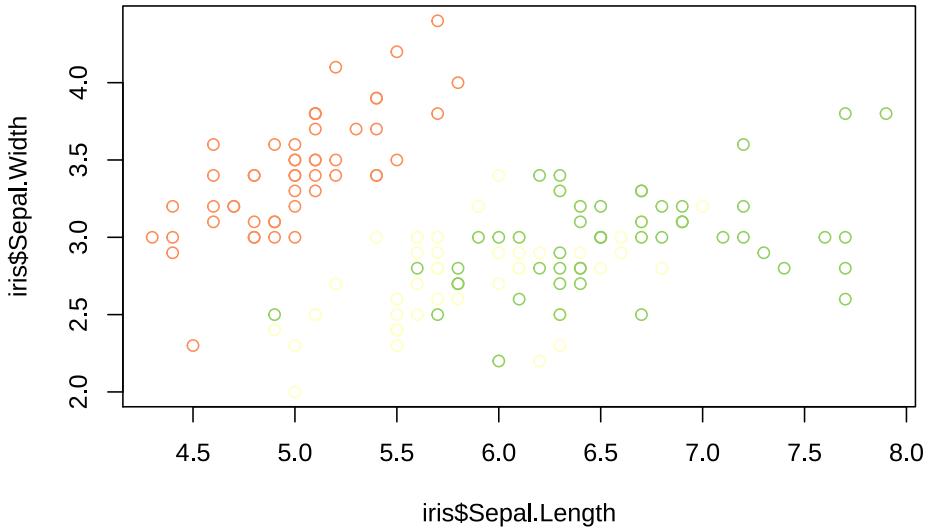
RColorBrewer 包提供了 3 套配色方案，分别为**连续型，极端型以及离散型**。

- **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。

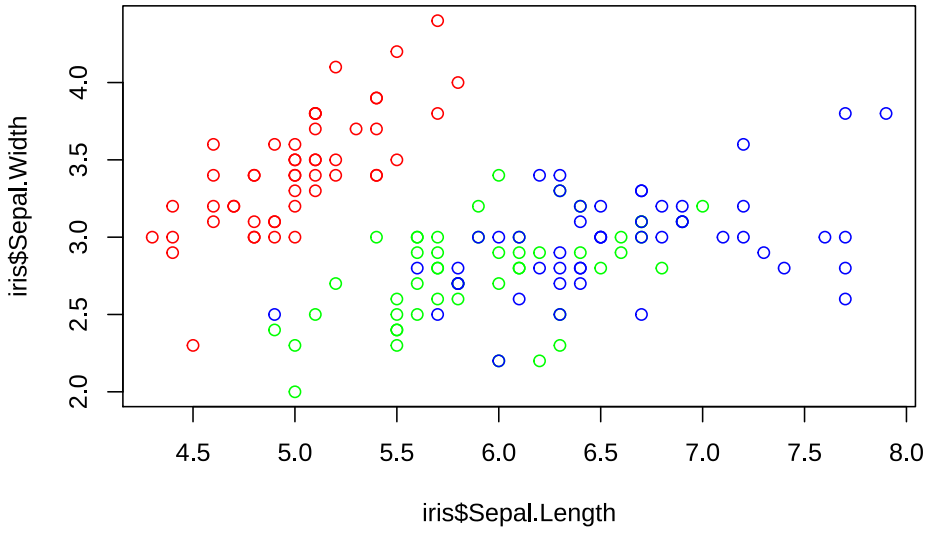
```
par(mfrow = c(1,3))
library(RColorBrewer)
par(mar=c(0.1,3,0.1,0.1))
display.brewer.all(type="seq")
display.brewer.all(type="div")
display.brewer.all(type="qual")
```




```
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))
```



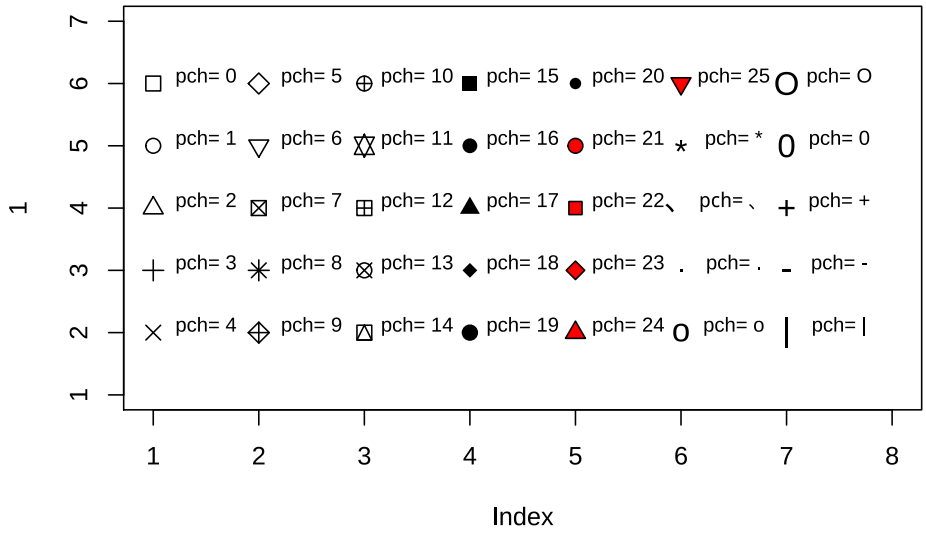

```
plot(iris$Sepal.Length, iris$Sepal.Width, col = rep(rainbow(3), each = 50))
```



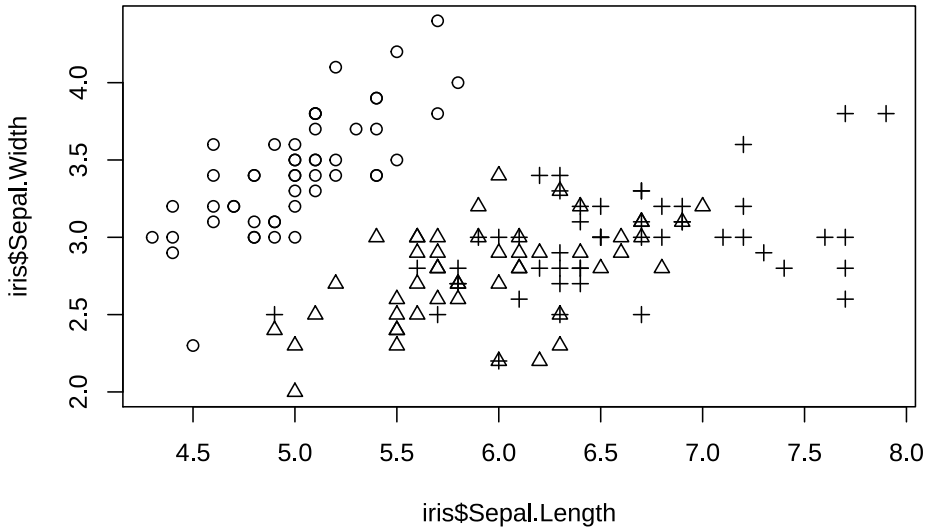
2.2 修改点符号与线条

2.2.1 点样式

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




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




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

2.2.2 线条样式

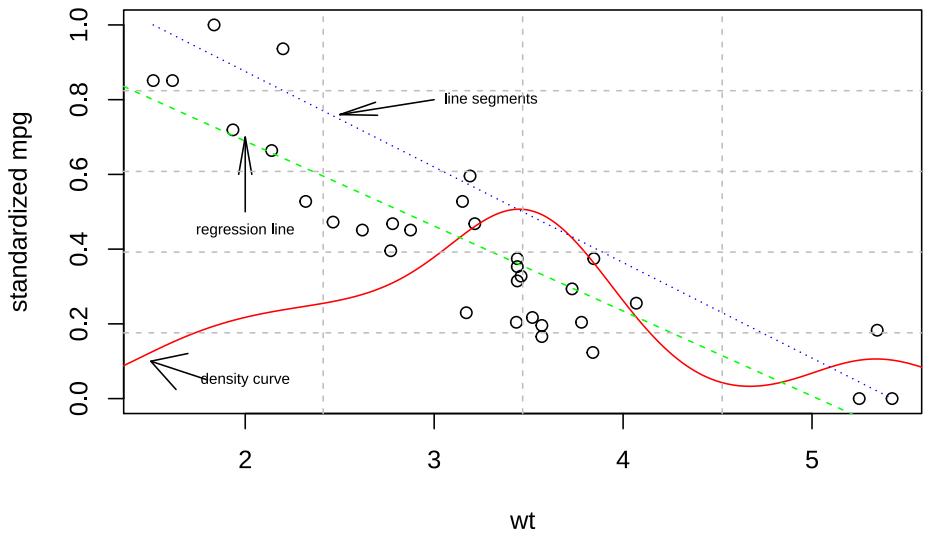
R 语言提供了绘制不同类别的线条的多种函数，主要有

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

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

以 mtcars 数据集为例来展示实际绘图过程中线条的应用。

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



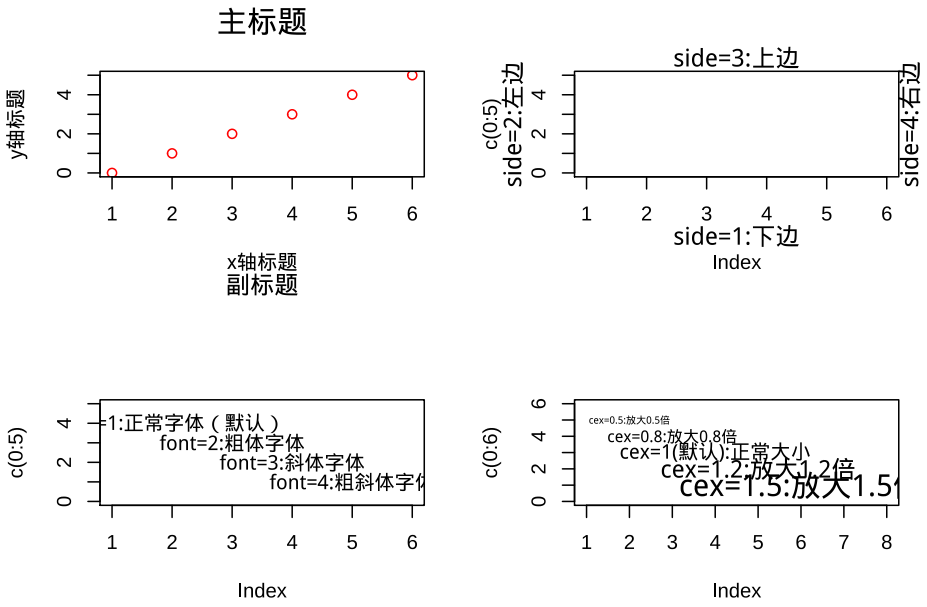
2.2.3 修改文本参数

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。

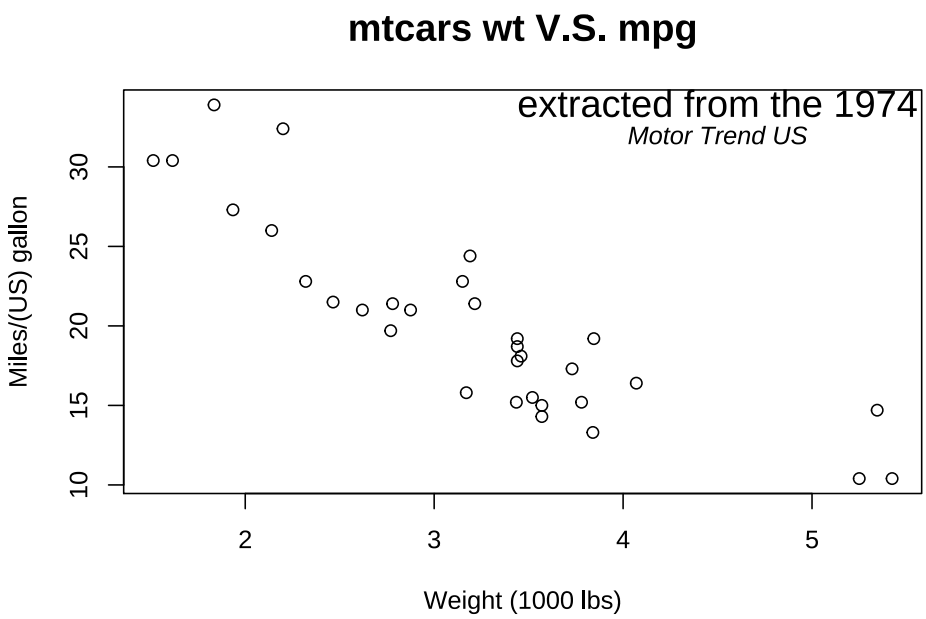
```
par(mfrow = c(2, 2))
# 图一: 图形添加标题
plot(c(0:5),col="red",xlab="",ylab="")
title(main=list(" 主标题",cex=1.5),sub=list(" 副标题",cex=1.2),
      xlab="x 轴标题",ylab="y 轴标题")
# 图二: 图形周边添加文本
plot(c(0:5),col="white")
mtext('side=1: 下边',side=1,line=2)
mtext('side=2: 左边' ,side=2,line=2)
mtext('side=3: 上边' ,side=3)
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
```



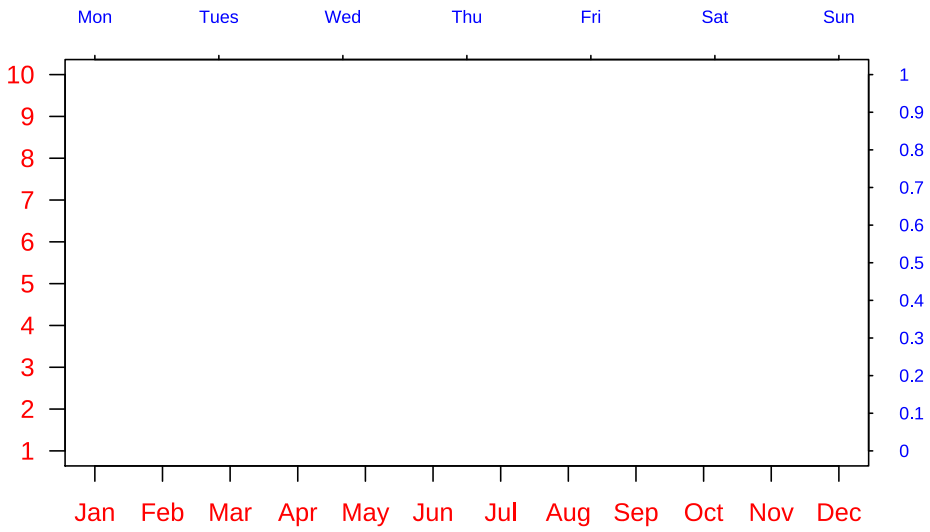
2.2.4 设置坐标轴

使用 `axis()` 进行设置坐标轴。

参数	描述
axes	逻辑参数，如果axes=TRUE（默认），则显示坐标轴，如果axes=FALSE，则隐藏坐标轴。
xaxt/yaxt	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
xaxs/yaxs	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
xlim/ylim	坐标轴范围，设置为c(from,to) , from是x /y轴的首坐标,to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,
```

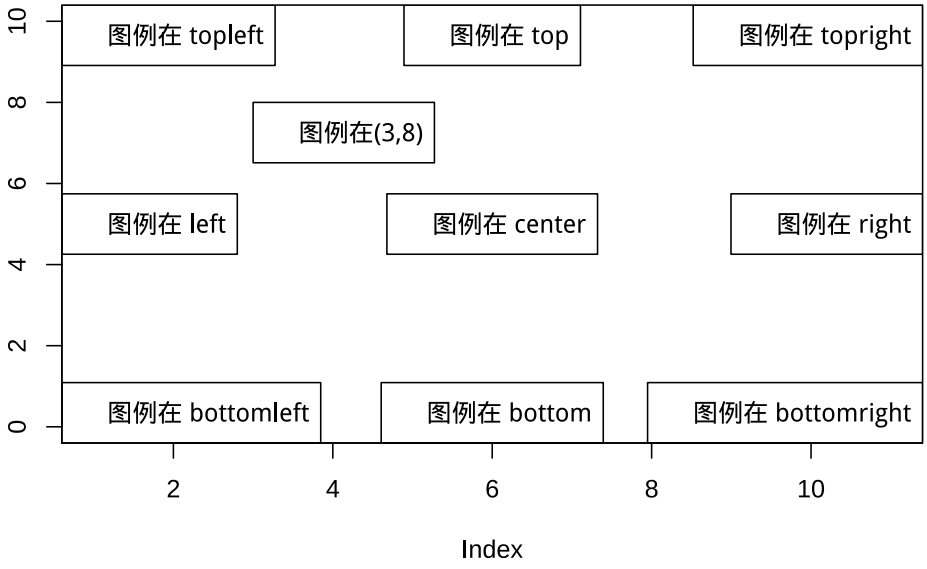
```
     tck=-0.01, labels = c("Mon", "Tues", "Wed", "Thu", "Fri", "Sat", "Sun"))
axis(4, at=seq(1,12,length=11), col.axis="blue", cex.axis=0.7,
     tck=-0.01, labels=seq(0, 1, 0.1), las=2)
```



2.2.5 添加图例

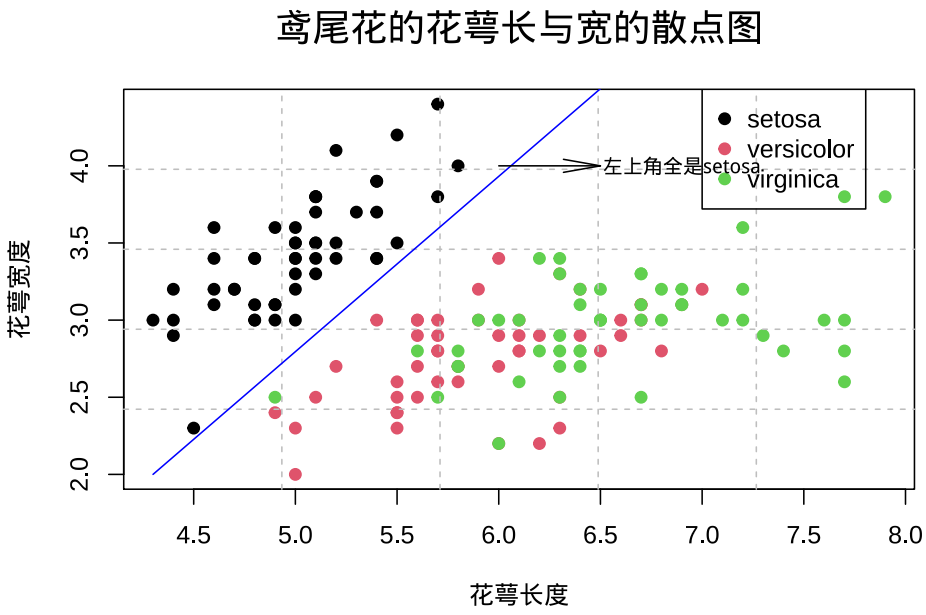
legend 函数的绘制图例的位置效果

```
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
```

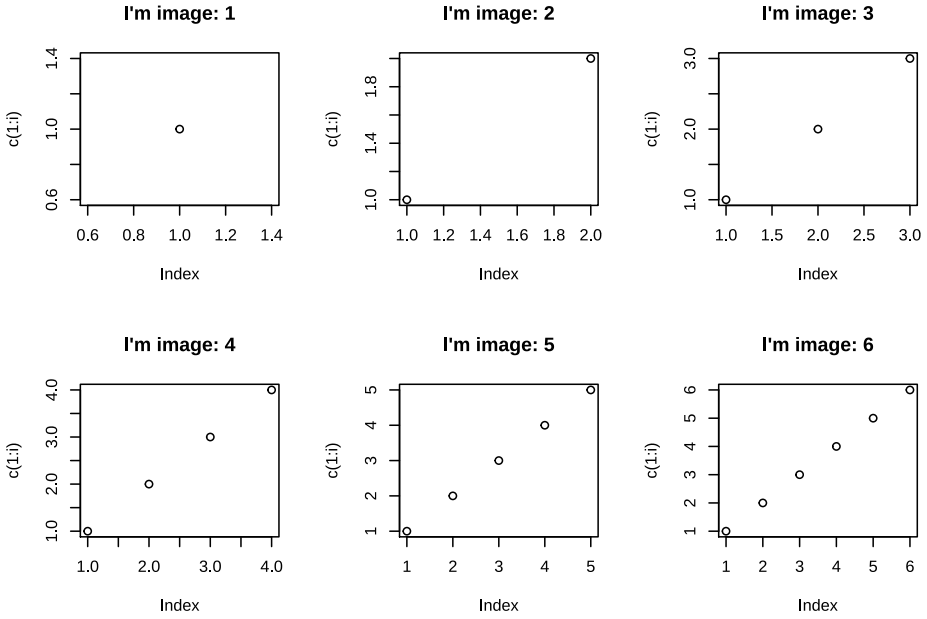


3 绘制组合图形

3.1 par()

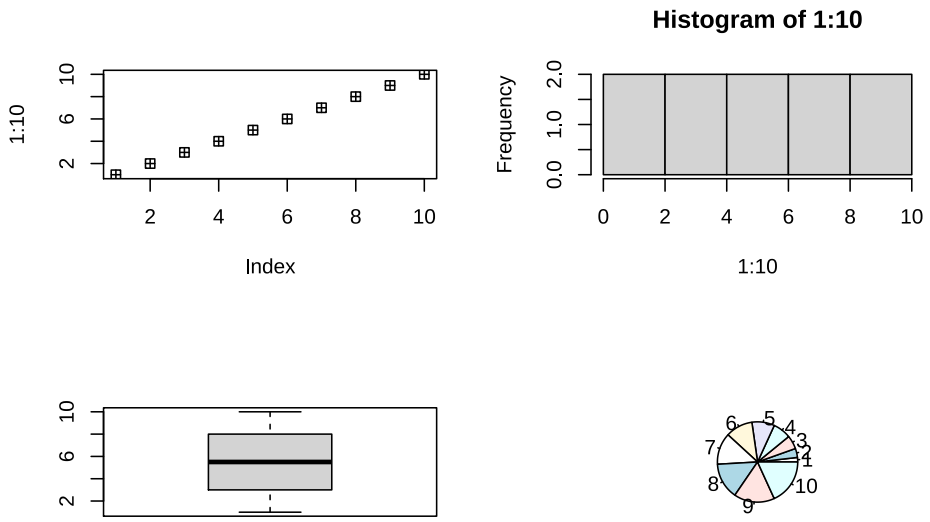
一页多图用 mfrow 参数或 mfcol 参数规定。

```
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(mfrow1)

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

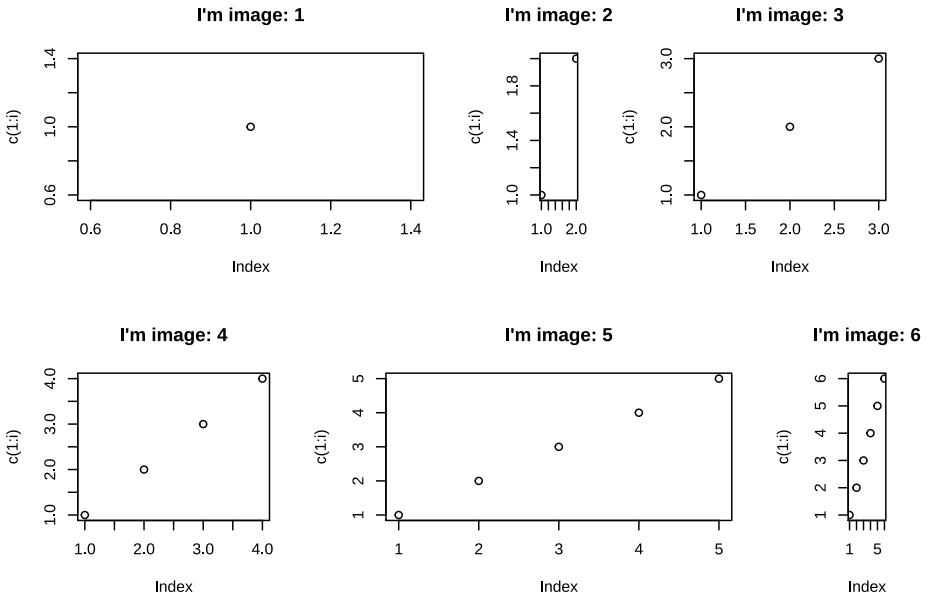



```
par(op)
```

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))
}
```



4 保存图形

4.1 使用代码

对于其他格式输出类似 pdf 的输出。

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

```
## pdf
##   2
```

4.2 在 Rstudio 窗口点击按钮保存

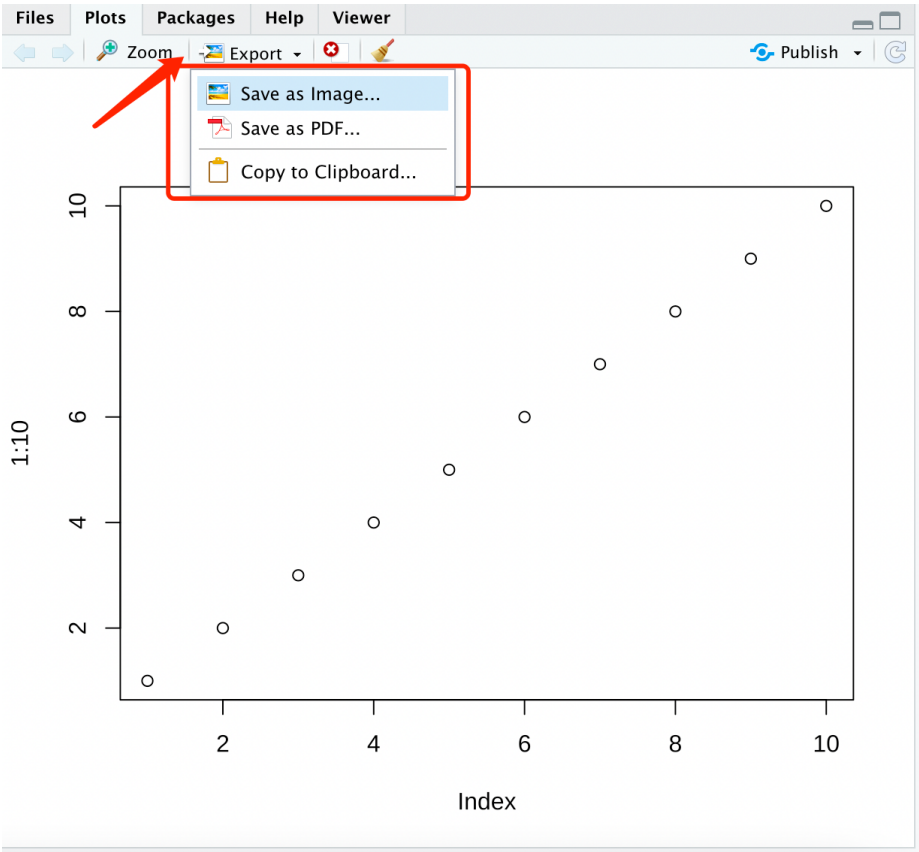


图 1: Rstudio 界面右下角

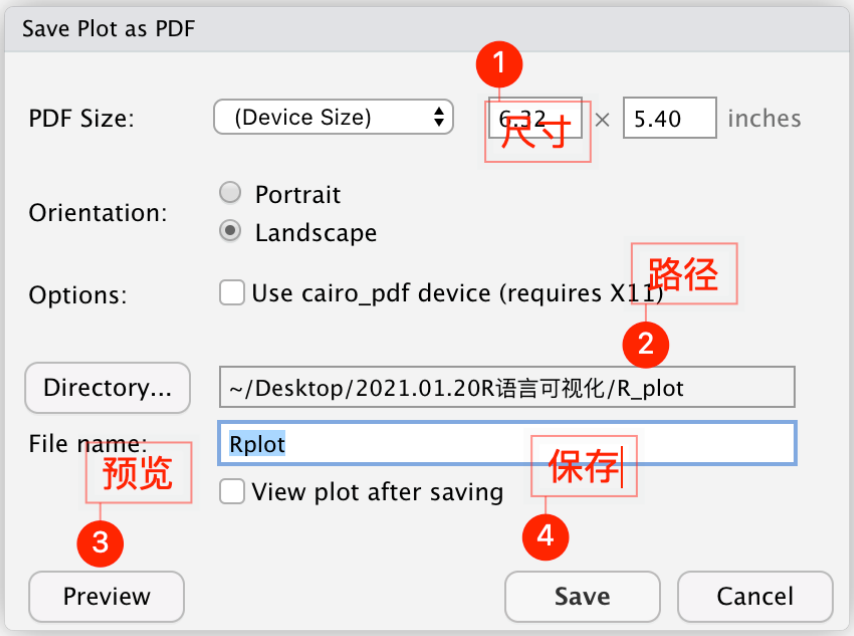

Save Plot as PDF
1
PDF Size:
(Device Size)
6.32
尺寸
×
5.40
inches
Orientation:
Portrait
Landscape
Options:
Use cairo_pdf device (requires X11)
路径
2
Directory...
~/Desktop/2021.01.20R语言可视化/R_plot
File name:
Rplot
预览
保存
View plot after saving
3
4
Preview
Save
Cancel

图 2: 自定义设置

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 3: 个人公众号：庄闪闪的 R 语言手册

其他联系方式（可跳转）：

- 知乎
- Github
- CSDN
- b 站

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。

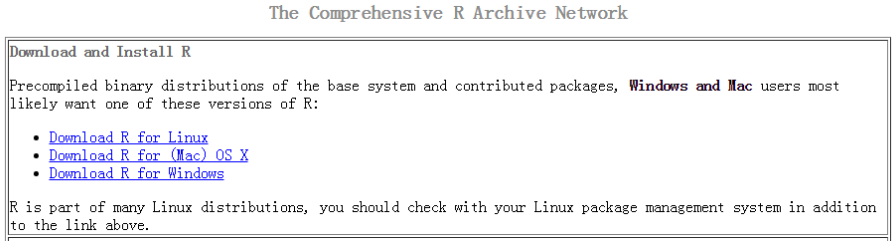
注：该部分来源于《数据科学中的 R 语言》。

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:



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	24ca3fe0dad8187aabd4bfbb9dc2b5ad
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	5e77094a88fdbddddb0d35708752462

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，

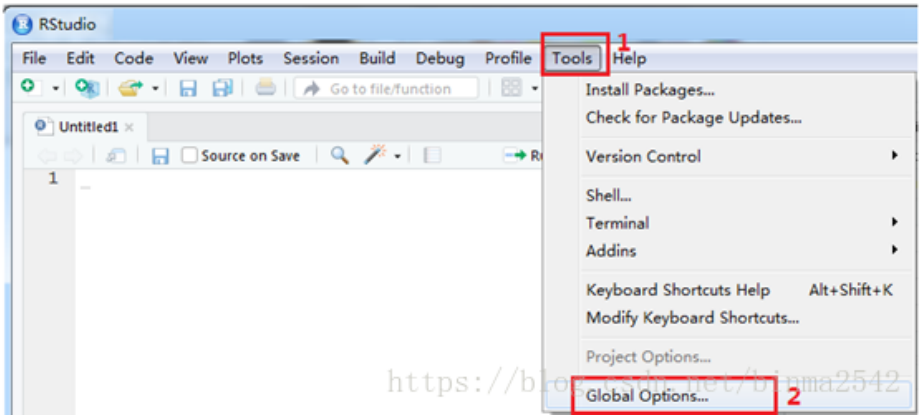


图 4: 选择国内镜像 1

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'
```

输入下面命令后，再试试

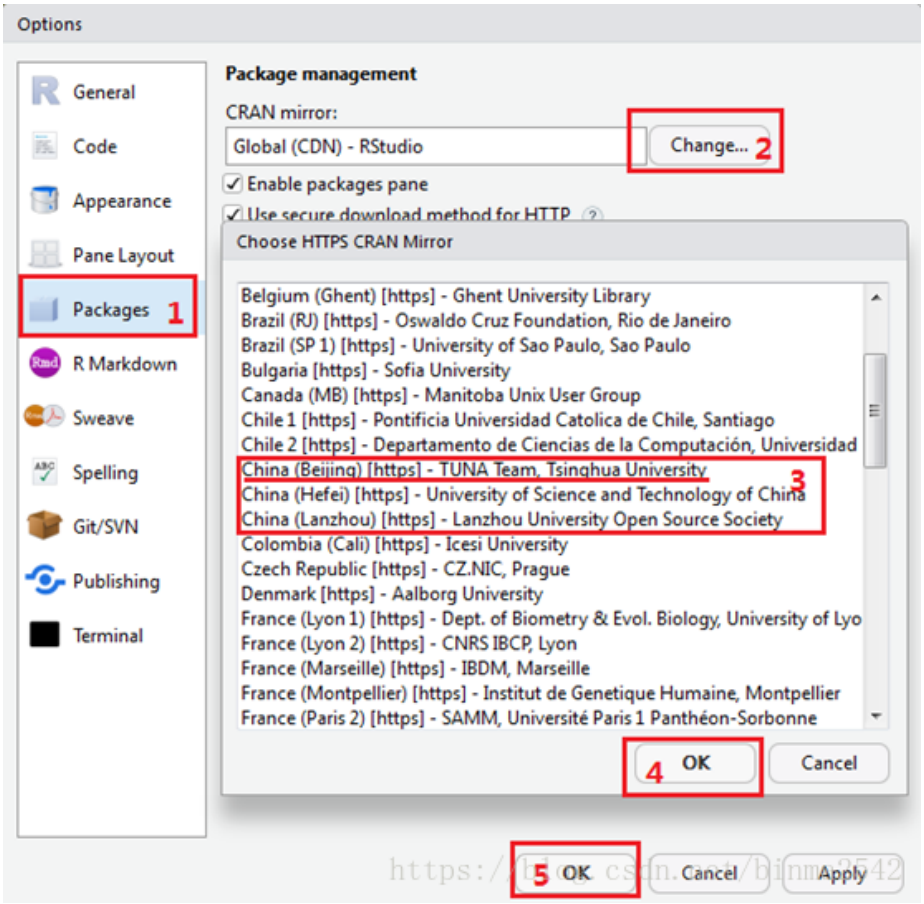

Options
General
Code
Appearance
Pane Layout
Packages 1
R Markdown
Sweave
Spelling
Git/SVN
Publishing
Terminal
Package management
CRAN mirror:
Global (CDN) - RStudio
Change... 2
Enable packages pane
Use secure download method for HTTP
Choose HTTPS CRAN Mirror
Belgium (Ghent) [https] - Ghent University Library
Brazil (RJ) [https] - Oswaldo Cruz Foundation, Rio de Janeiro
Brazil (SP 1) [https] - University of Sao Paulo, Sao Paulo
Bulgaria [https] - Sofia University
Canada (MB) [https] - Manitoba Unix User Group
Chile 1 [https] - Pontificia Universidad Catolica de Chile, Santiago
Chile 2 [https] - Departamento de Ciencias de la Computación, Universidad
China (Beijing) [https] - TUNA Team, Tsinghua University
China (Hefei) [https] - University of Science and Technology of China
China (Lanzhou) [https] - Lanzhou University Open Source Society
3
Colombia (Cali) [https] - Icesi University
Czech Republic [https] - CZ.NIC, Prague
Denmark [https] - Aalborg University
France (Lyon 1) [https] - Dept. of Biometry & Evol. Biology, University of Lyo
France (Lyon 2) [https] - CNRS IBCP, Lyon
France (Marseille) [https] - IBDM, Marseille
France (Montpellier) [https] - Institut de Genetique Humaine, Montpellier
France (Paris 2) [https] - SAMM, Université Paris 1 Panthéon-Sorbonne
4 OK
Cancel
5 OK
Cancel
Apply

图 5: 选择国内镜像 2

```
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

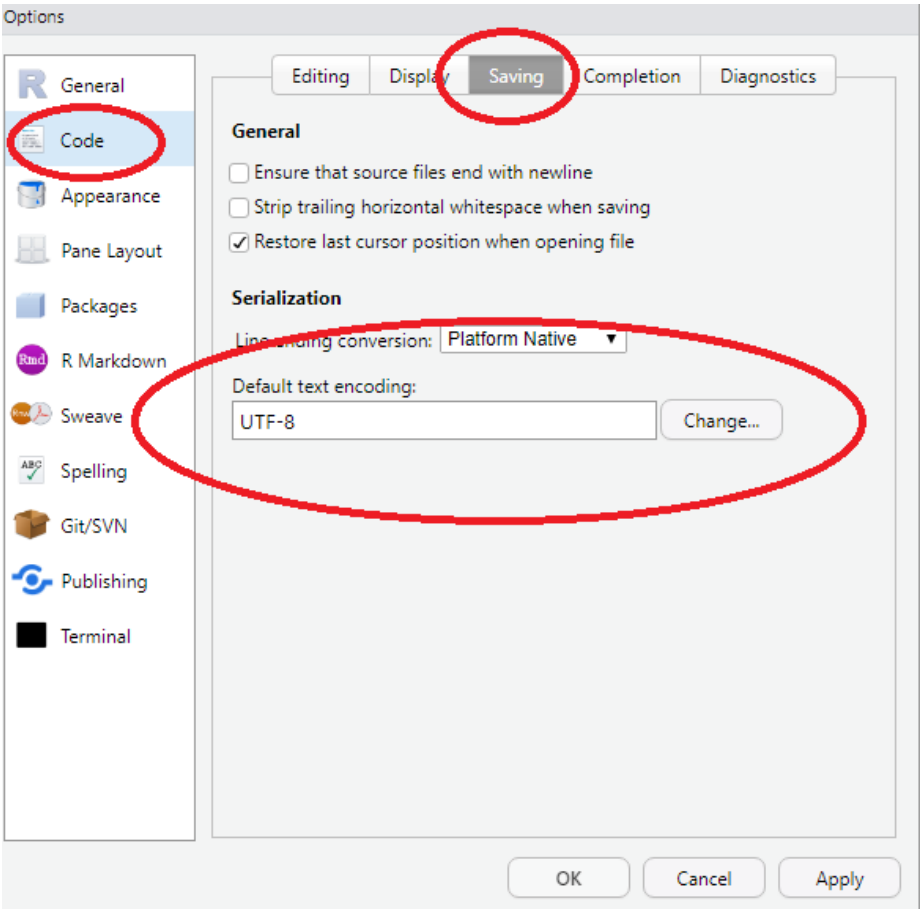


图 6: 代码是乱码

- 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

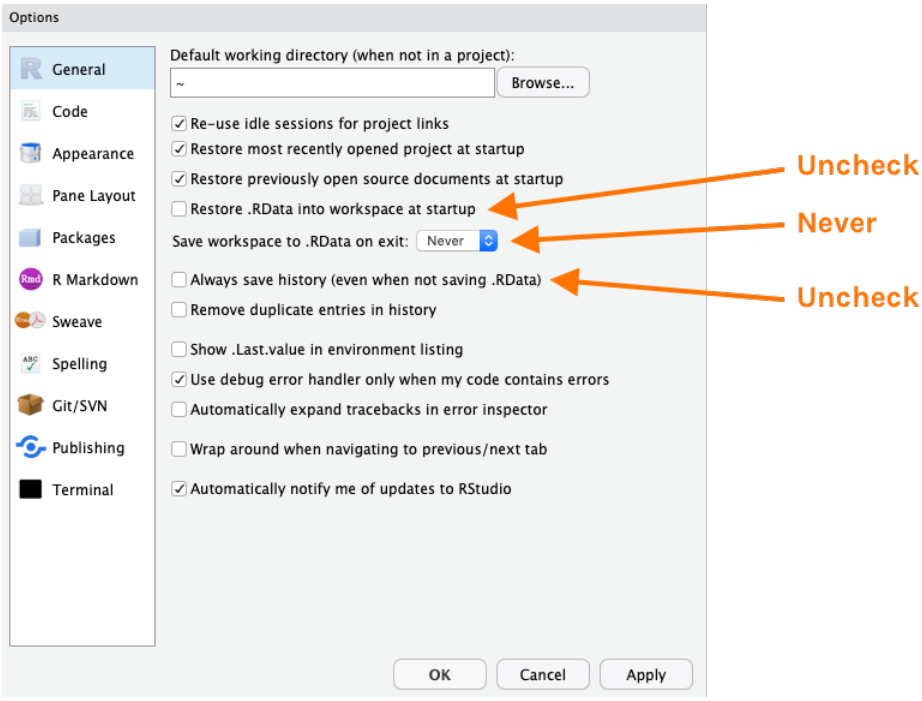


图 7: 打开 Rstudio 非常慢

- 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

- 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：

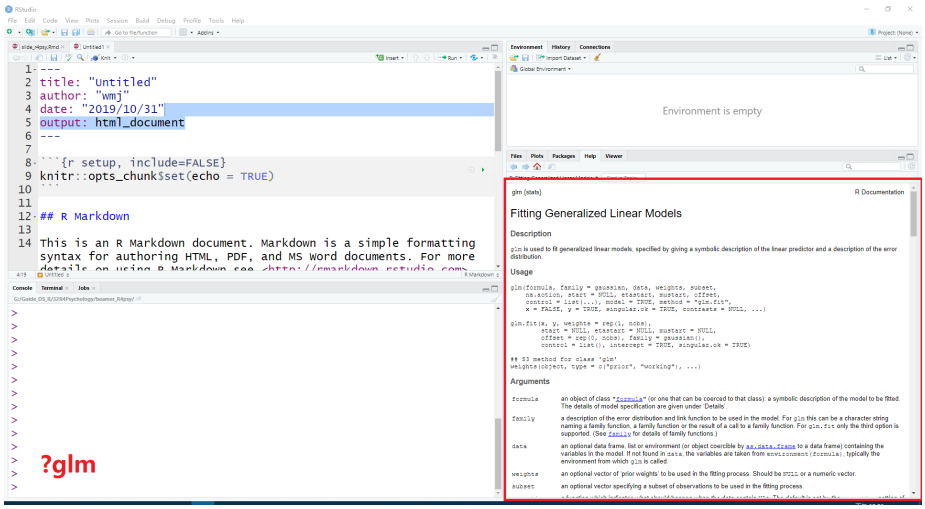


图 8: help 帮助

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/