心理统计

战立侃

2018 年 10 月 10 日

目录

第一章　数据探索

1.1　什么是统计

1.2　什么是数据

数据（data）、数据类型：质量型（qualitative）和数量型（quantitative）
数量型：非连续型（discrete）和连续型（continuous）；

测量类型：称名类（nominal）、顺序类（ordinal）、等距型（interval）、等比型（ratio）

1.3　定性数据

1.3.1　频率分析表

R 语言中画频率分析表的两个常见函数为 `table()` 和 `xtabs()`。例如

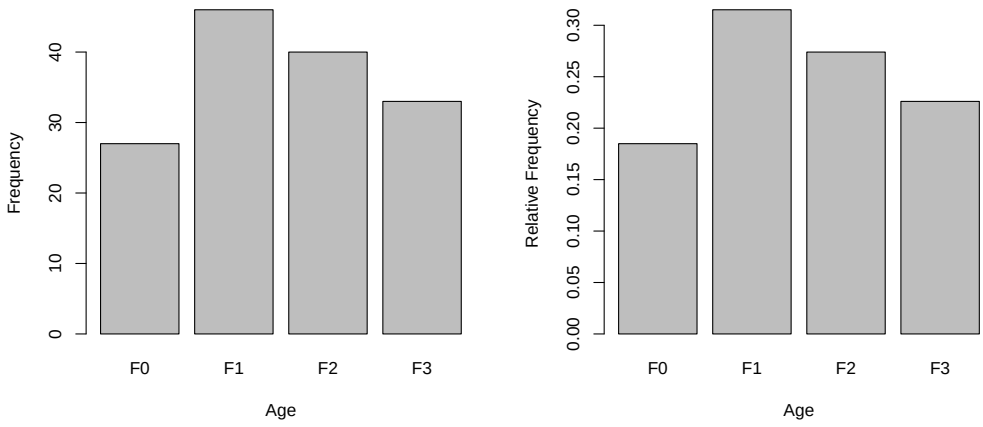
```
library(MASS)
with(data = quine, table(Age))
table(quine[, "Age"]) / length(quine[, "Age"])
prop.table(table(quine[, "Age"]))
xtabs( ~ Age, data = quine)

Age
```

```
F0 F1 F2 F3
27 46 40 33

       F0        F1        F2        F3
0.1849315 0.3150685 0.2739726 0.2260274

       F0        F1        F2        F3
0.1849315 0.3150685 0.2739726 0.2260274
Age
F0 F1 F2 F3
27 46 40 33
```

1.3.2 柱形图

柱形图函数 barplot() 的输入数据应该是总结后的频率分析表数据。例如

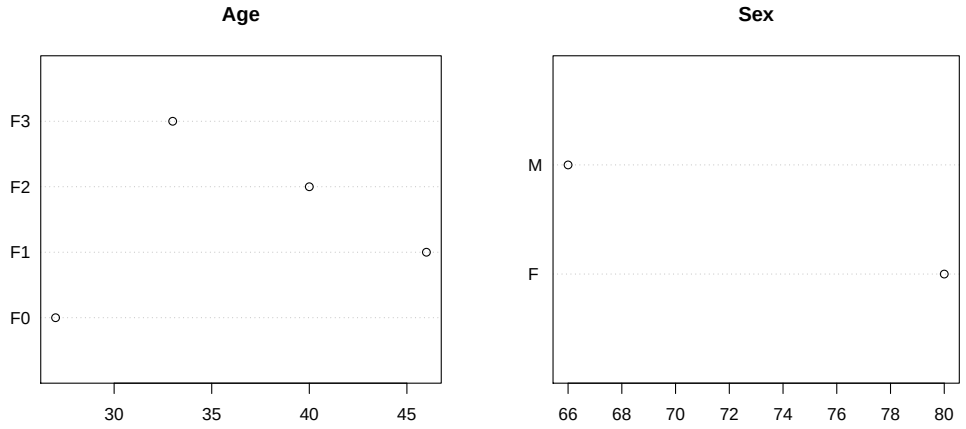
```
par(mfrow = c(1, 2))
tb1 <- xtabs( ~ Age, data = quine)
tb2 <- prop.table(table(quine[, "Age"]))
barplot(tb1, xlab = "Age", ylab = "Frequency")
barplot(tb2, xlab = "Age", ylab = "Relative Frequency")
```



1.3.3 点状图

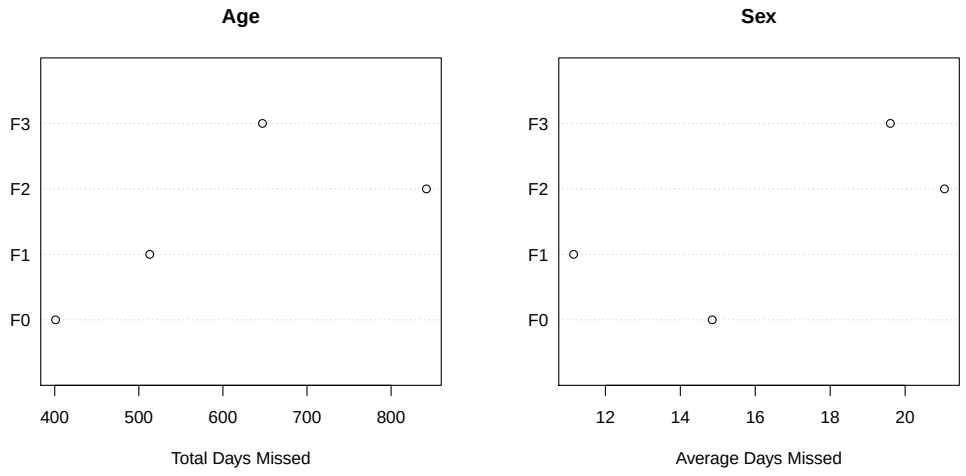
频率数据的另一种展示方式是点状图（Dot charts）

```
tb3 <- xtabs( ~ Sex, data = quine)
par(mfrow = c(1, 2))
dotchart(tb1, main = "Age")
dotchart(tb3, main = "Sex")
```



点状图也可以用来画二维数据，例如

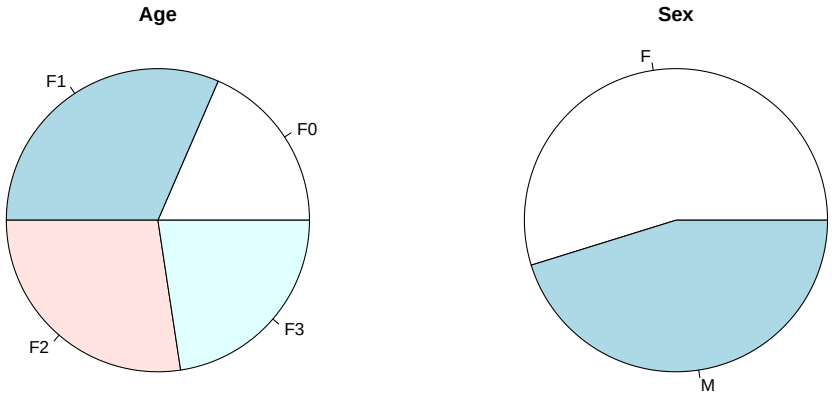
```
tb4 <- xtabs(Days ~ Age, data = quine)
tb5 <- with(quine, tapply(Days, list(Age), mean))
par(mfrow = c(1, 2))
dotchart(tb4, main = "Age", xlab = "Total Days Missed")
dotchart(tb5, main = "Sex", xlab = "Average Days Missed")
```



1.3.4 饼状图

饼状图（Pie charts）展示是不同类别的相对频率或百分比。

```
tb3 <- xtabs( ~ Sex, data = quine)
par(mfrow = c(1, 2))
pie(tb1, radius = 1, main = "Age")
pie(tb3, radius = 1, main = "Sex")
```



1.4 定量数据

1.4.1 茎叶图

茎叶图（Stem-and-Leaf Plots）是连续型数据的一种展示方式

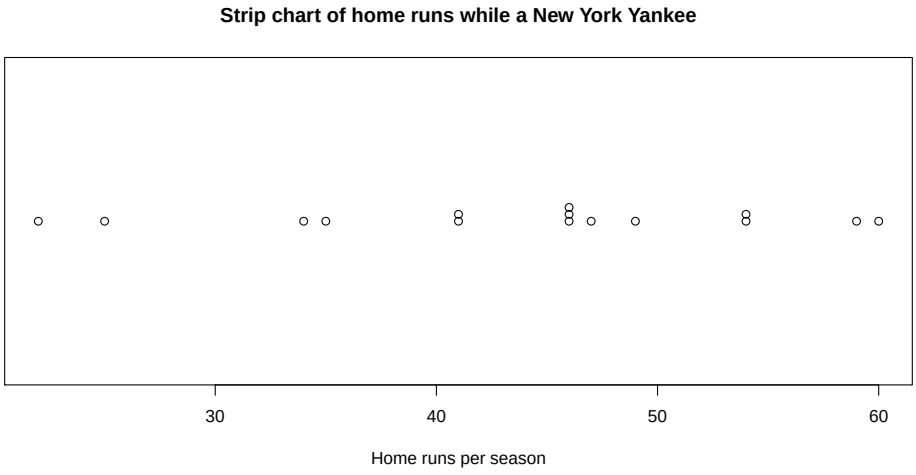
```
NYYHR <- PASWR::Baberuth[, "HR"][
  PASWR::Baberuth[, "Team"] == "NY-A"]
stem(NYYHR)
```

```
  The decimal point is 1 digit(s) to the right of the |

  2 | 25
  3 | 45
  4 | 1166679
  5 | 449
  6 | 0
```

1.4.2 散点图

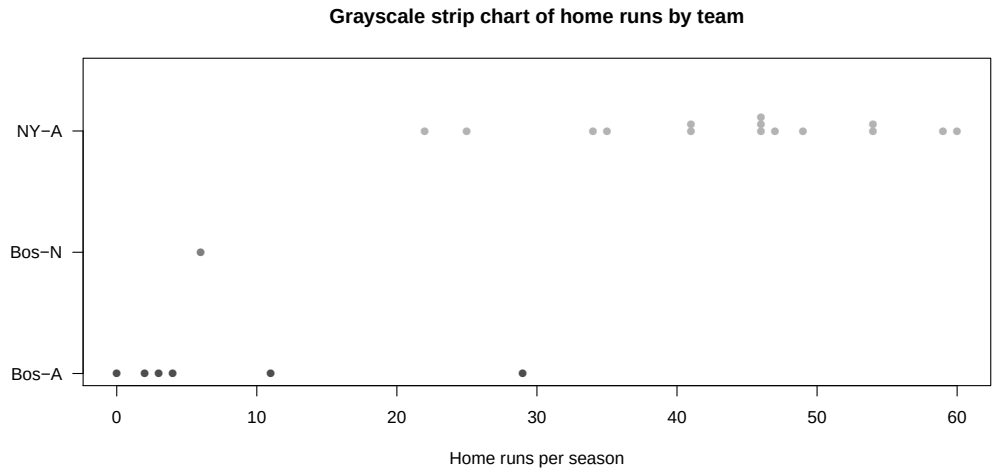
散点图 (stripchart) 中的方法改为 `method = "stack"`:

```
NYYHR <- PASWR::Baberuth[, "HR"][7:21]
stripchart(NYYHR, method = "stack", pch = 1,
  main = "Strip chart of home runs while a New York Yankee",
  xlab = "Home runs per season")
```



散点图也可以用来表示两个变量之间的关系

```
par(las = 1) # 调转纵坐标标签
stripchart(HR ~ Team, data = PASWR::Baberuth, method = "stack",
  pch = 16, col = paste("gray", 30 + (0:2) * 20, sep = ""),
  xlab = "Home runs per season",
  main = "Grayscale strip chart of home runs by team")
```



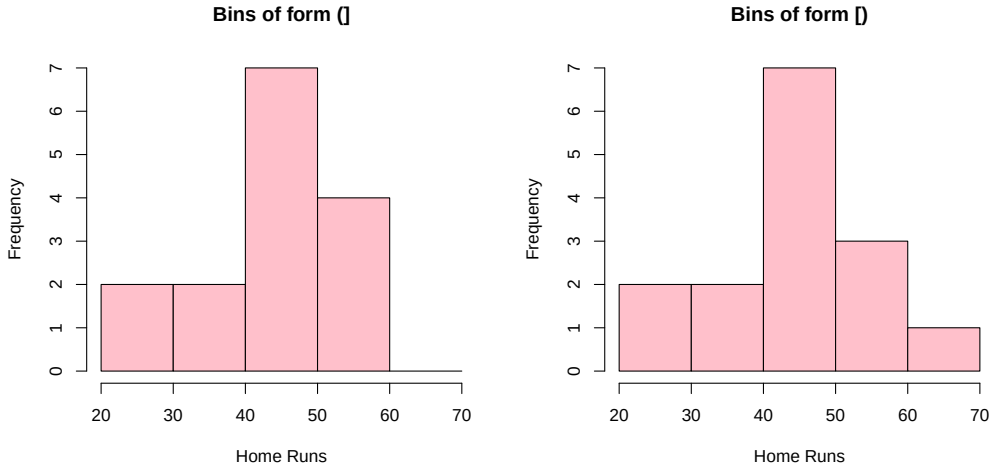
1.4.3 密度曲线

1.4.3.1 直方图

直方图 (histograms) 函数 `hist()`。默认情况下，直方图计算频率区间是左开右闭(]。我们也可以把频率计算的区间改为左闭右开[)，即设定如下参数 `right = FASE`。当区间为左闭右开时，直方图的结果和茎叶图类似。例如：

```
opar <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
bin <- seq(from = 20, to = 70, by = 10)
hs1 <- hist(NYYHR, breaks = bin,
```

```
  xlab = "Home Runs", col = "pink", main = "Bins of form (]")
hs2 <- hist(NYYHR, breaks = bin, right = FALSE,
  xlab = "Home Runs", col = "pink", main = "Bins of form [)")
```

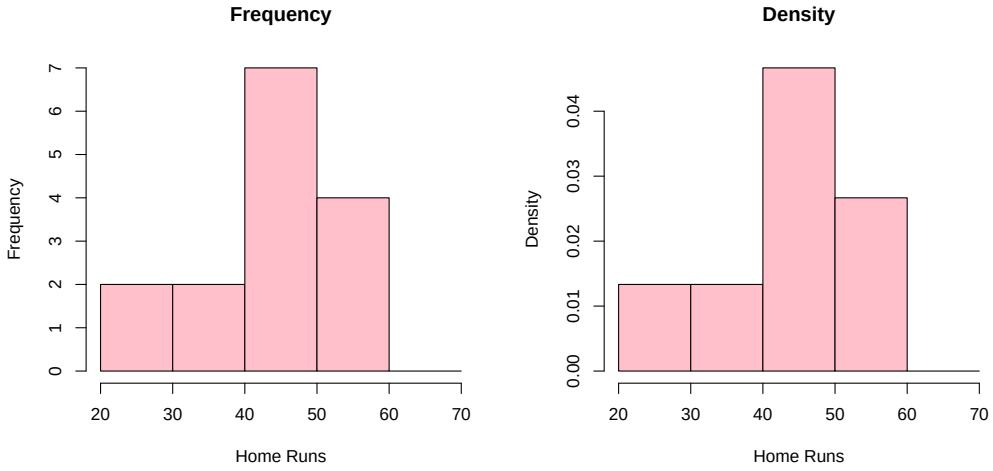


直方图的默认画的是数据的绝对频率。我们也可以把直方图画为相对频率，即概率密度。如果我们要把一个样本量为 n 的数据划分为区间宽度为 h 组别，那么每个组别的概率密度可以用公式 1.1 计算：

$$\hat{f}(x) = \frac{v_k}{nh}, \quad t_k < x \leq t_{k+1} \tag{1.1}$$

此处 v_k 指落入类别区间 $(t_k, t_{k+1}]$ 的样本数据点的个数。

```
opar <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
hs1 <- hist(NYYHR, breaks = bin,
  xlab = "Home Runs", col = "pink", main = "Frequency")
hs4 <- hist(NYYHR, breaks = bin, freq = FALSE,
  xlab = "Home Runs", col = "pink", main = "Density")
```



除了通过一个数组（vector）人为的确定直方图的分界点外，R 语言还内置了一些计算方法。这些方法也是通过参数 breaks= 来设定的。该参数可选方法有三种："Sturges"，"FD"/"Freedman-Diaconis"，和 "Scott"，其中默认的方法是第一种 breaks="Sturges"。

- Sturges 计算分组宽度的公式为 1.2

$$h_{\text{Sturges}} = \frac{R}{1 + \log_2 n} \tag{1.2}$$

其中 R 是样本的全距（Range）。该方法假定数据服从正态分布，所以如果数据为偏态或多峰分布，那么用该方法确定区间宽度就是不合适的。下面是一个例子：

```
(xs <- PASWR::Baberuth[, "HR"][7:21]) # 待分析数据
 R <- diff(range(xs))                 # 计算数据的全距(Range)
 n <- length(xs)                      # 计算样本容量(数据个数)
(hs <- R / (1 + log2(n)))             # 计算分组宽度
(nclassS <- ceiling(R / hs))          # 根据分组宽度计算分组个数
```

```
(nclassS <- nclass.Sturges(xs))    # R计算分组个数的方法
                                   #（与手动计算的结果相同）
(bpS <- min(xs) + hs * 0:nclassS) # 根据分组个数计算分组位置
(bpSp <- pretty(xs, n = nclassS)) # pretty() 计算的分组位置
 hs5 <- hist(xs, breaks = "Sturges", plot = FALSE)
 hs5[["breaks"]]                   # hist 实际使用的分组位置

 [1] 54 59 35 41 46 25 47 60 54 46 49 46 41 34 22
[1] 7.744212
[1] 5
[1] 5
[1] 22.00000 29.74421 37.48842 45.23264 52.97685 60.72106
[1] 20 30 40 50 60
[1] 20 30 40 50 60
```

- Freedman-Diaconis 计算分组宽度的公式为 1.3:

$$h_{\text{FD}} = \frac{2 \cdot (IQR)}{n^{1/3}} \tag{1.3}$$

其中 IQR 指的是数据的四分位距（interquartile range）。下面是一个例子:

```
(xs <- PASWR::Baberuth[, "HR"][7:21]) # 待分析数据
 n <- length(xs)                      # 计算样本容量(数据个数)
(hfd <- 2 * IQR(xs) / (n ^ (1 / 3)))# 计算分组宽度
(nclassFD <- ceiling(R / hfd))       # 根据分组宽度计算分组个数
(nclassFD <- nclass.FD(xs))    # R计算分组个数的方法
                               #（与手动计算的结果相同）
```

```
(bpFD <- min(xs) + hfd * 0:nclassFD) # 根据分组个数计算分组位置
(bpFDp <- pretty(xs, n = nclassFD)) # pretty() 计算的分组位置
 hs6 <- hist(xs, breaks = "FD", plot = FALSE)
 hs6[["breaks"]]                    # hist 实际使用的分组位置

 [1] 54 59 35 41 46 25 47 60 54 46 49 46 41 34 22
[1] 10.94796
[1] 4
[1] 4
[1] 22.00000 32.94796 43.89593 54.84389 65.79185
[1] 20 30 40 50 60
[1] 20 30 40 50 60
```

在使用该方法时，虽然根据公式计算出的结果不一样。但是经由 pretty() 函数修正后结果第一种方法是一样的。

- Scott 计算区间宽度的方法为 1.4：

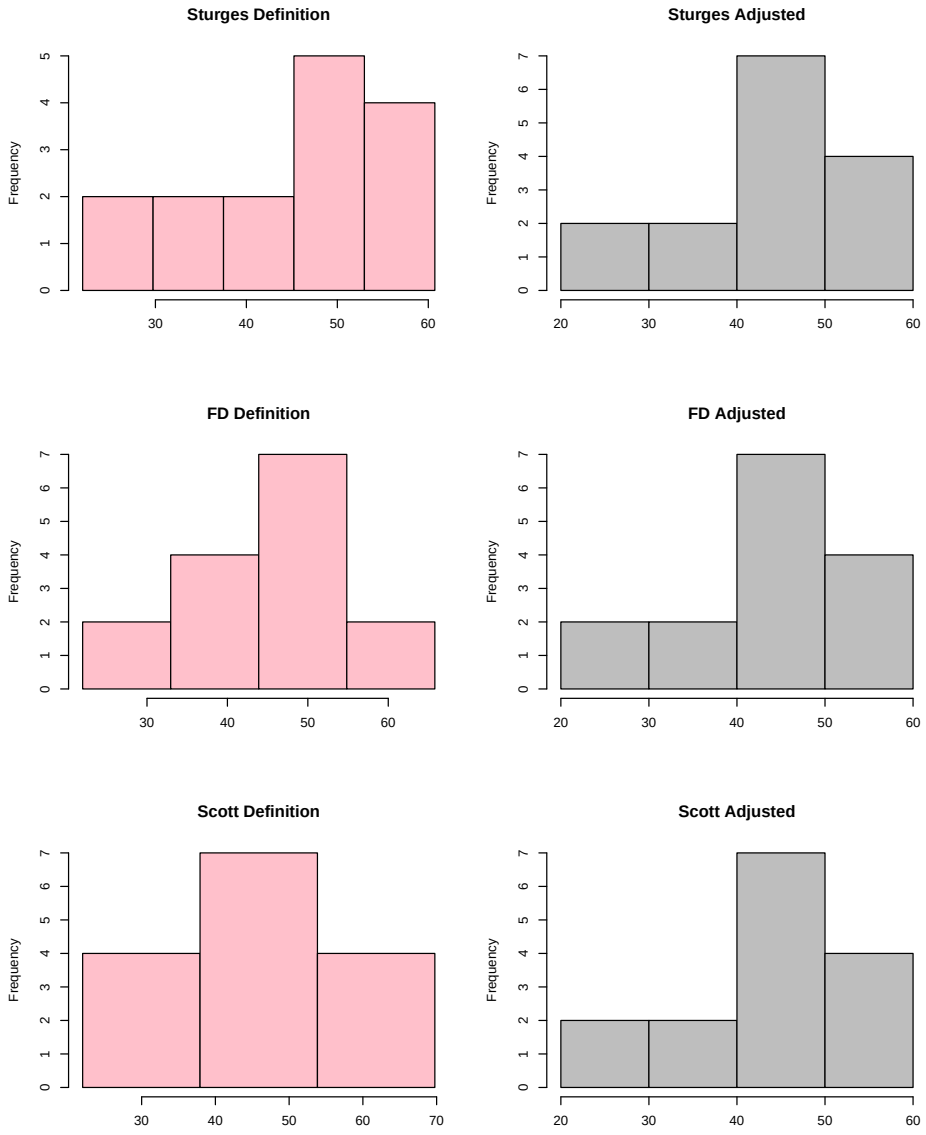
$$
\begin{aligned}
h_{\text{Scott}} &= \frac{2 \cdot 3^{1/3} \cdot \pi^{1/6} \cdot \hat{\sigma}}{n^{1/3}} \\
h_{\text{ScottR}} &= \frac{3.5 \cdot \hat{\sigma}}{n^{1/3}}
\end{aligned}
\tag{1.4}
$$

其中 $\hat{\sigma}$ 是整体标准差的估计值。当然 R 并不直接使用上述公式计算出来的值，而是用函数 pretty() 对分组宽度进一步优化。

```
(xs <- PASWR::Baberuth[, "HR"][7:21])# 待分析数据
 n <- length(xs)                     # 计算样本容量(数据个数)
(hsc <- 2 * 3 ^ (1 / 3) * pi ^(1 / 6) * sd(xs) / n ^ (1 / 3))
                                     # 计算分组宽度
```

```
(hscR <- 3.5 * sd(xs) / n ^ (1 / 3)) # 计算分组宽度
(nclassSC <- ceiling(R / hsc))       # 根据分组宽度计算分组个数
(nclassSC <- nclass.scott(xs))       # R计算分组个数的方法
                                     #(与手动计算的结果相同)
(bpSC <- min(xs) + hsc * 0:nclassSC) # 根据分组个数计算分组位置
(bpSCp <- pretty(xs, n = nclassSC))  # pretty() 计算的分组位置
 hs7 <- hist(xs, breaks = "Scott", plot = FALSE)
 hs7[["breaks"]]                     # hist 实际使用的分组位置

 [1] 54 59 35 41 46 25 47 60 54 46 49 46 41 34 22
[1] 15.91972
[1] 15.96154
[1] 3
[1] 3
[1] 22.00000 37.91972 53.83944 69.75916
[1] 20 30 40 50 60
[1] 20 30 40 50 60
```

下面是六幅图的左右两侧分别是根据定义画出来的图和用pretty()函数修正后的图:


Sturges Definition
Frequency
0 1 2 3 4 5
30 40 50 60
Sturges Adjusted
Frequency
0 1 2 3 4 5 6 7
20 30 40 50 60
FD Definition
Frequency
0 1 2 3 4 5 6 7
30 40 50 60
FD Adjusted
Frequency
0 1 2 3 4 5 6 7
20 30 40 50 60
Scott Definition
Frequency
0 1 2 3 4 5 6 7
30 40 50 60 70
Scott Adjusted
Frequency
0 1 2 3 4 5 6 7
20 30 40 50 60

1.4.3.2 核密度估计

核概率密度估计（kernel density estimator）是直方图密度的扩展，其计算公式如下：

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1.5}$$

其中 $K(\cdot)$ 是核函数（kernel function），h 是平滑参数 (smoothing parameter) 或组别宽度 (bandwidth)。核函数可以是满足以下条件的任何一个对称性的概率密度函数：

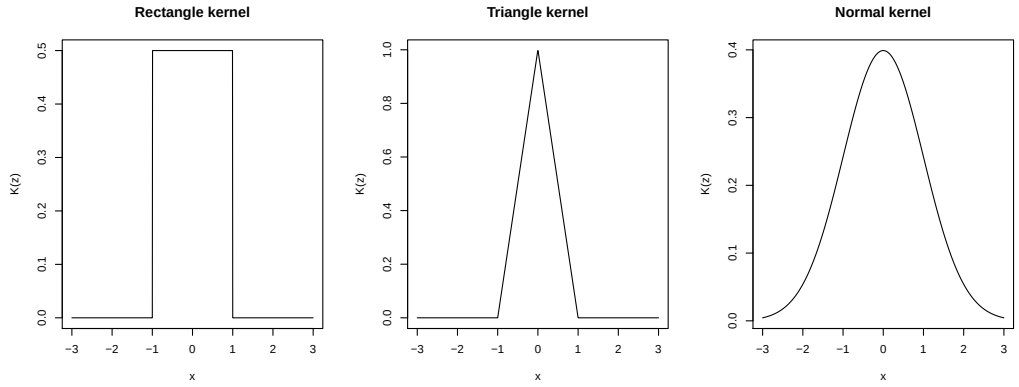
$$\int_{\prime\prime !}^{!} K(x)dx = 1$$

其中常见的核函数有长方形分布核（rectangular kernel）、三角形分布核（triangular kernel）和正态分布核（normal 或 Gaussian）。其定义为表 1.1：

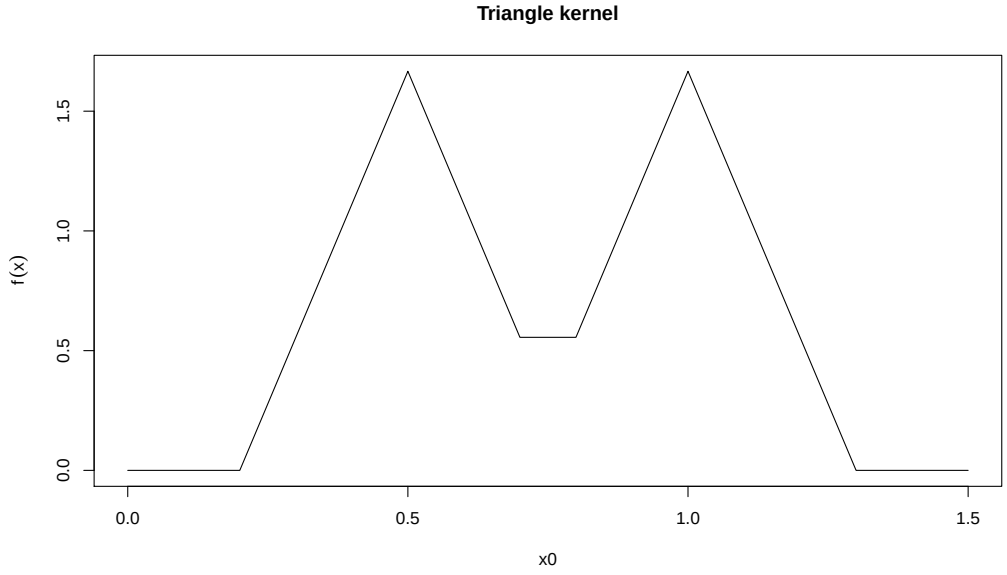
名称	定义
长方形	$K(x) = \frac{1}{2}, \quad \lvert x\rvert < 1$
三角形	$K(x) = 1 - \lvert x\rvert, \quad \lvert x\rvert < 1$
正态分布	$K(x) = \$\frac{1}{2\pi} e^{\frac{1}{2}x^2}, \quad -\infty < x < +\infty$

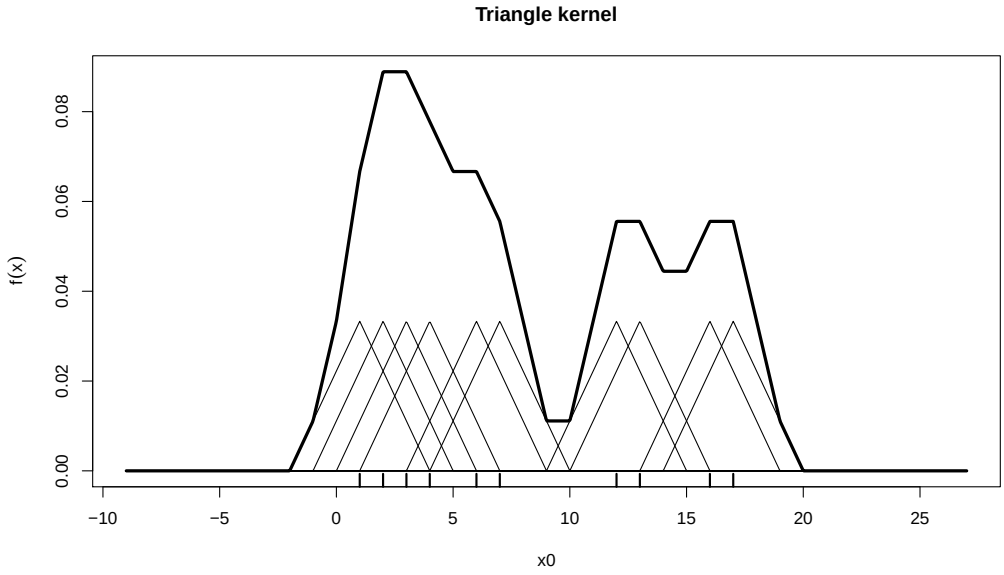
表 1.1: 常见核密度函数的定义

当区组宽度为 1 时，不同核密度函数对 x 值的加权作用可以用下图表示



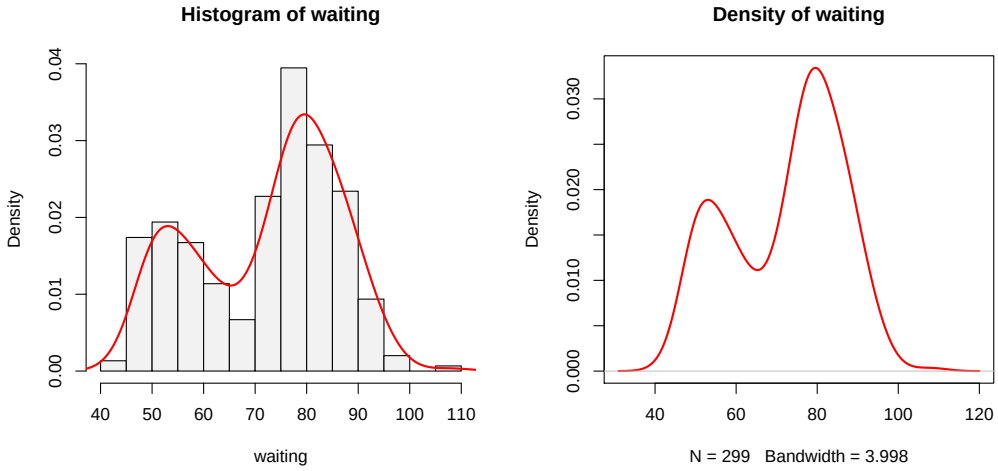
假如我们有两个数据 $0.5, 1.0$。我们把





R 语言中有一个函数叫 density()。其使用方法如下：

```
waiting <- MASS::geyser[, "waiting"]
density <- density(waiting)
par(mfrow = c(1, 2))
hist(waiting, freq = FALSE, col = "grey95")
lines(density, col = "red", lwd = 2)
plot(density(waiting), col = "red", lwd = 2,
  main = "Density of waiting")
```



它基本原理跟我们介绍的一样，其使用发发如下但计算更复杂。`density()` 允许的核密度函数更多，包括 `guassian`, `epanechnikov`, `rectangular`, `triangular`, `biweight`, `cosine`, 和 `optcosine`。

1.5 位置测量

1.5.1 平均值

一个容量为 n 的样本 $x_1, x_2, \cdots, x_n$ 的平均值（sample mean）$\overline{x}$ 的计算公式 1.6:

$$\overline{x} = \frac{x_1, x_2, \cdots, x_n}{n} = \sum_{i=1}^{n} \frac{x_i}{n} \tag{1.6}$$

```
(xs <- PASWR::Baberuth[, "HR"][7:21]) # 待分析数据
Mn <- function(x) sum(x) / length(x)
Mn(xs)

 [1] 54 59 35 41 46 25 47 60 54 46 49 46 41 34 22
```

```
[1] 43.93333
```

R 语言中平均值的函数为 mean()。其中参数 na.rm = TRUE 指是否把缺失值删除掉；参数 trim = p 指把数据从小到大排列后从两端各删除 $p\%$ 的数据，即 $p*n$ 个数。例如：

```
p <- 0.10
pp <- floor(p * length(xs))
xs.trim <- sort(xs)[(1 + pp):(length(xs) - pp)]
mean(xs)
mean(xs, trim = 0.10)
mean(xs.trim)

[1] 43.93333
[1] 44.38462
[1] 44.38462
```

此例子中，原始数据的平均值为 43.9333。当两端各删除掉 10% 的数据后，新平均值就变成了 44.3846。

1.5.2 中数

中数(sample median)的计算方法如下：首先把样本数据 $x_1, x_2, \cdots, x_n$ 从小到大排列，形成顺序统计量 (order statistics)，记为：$x_{(1)}, x_{(2)}, \cdots, x_{(n)}$。此时中数可以用如下公式计算：

$$m = \begin{cases} x_{(k+1)} & n = 2k+1(odd), \\ \frac{1}{2}(x_{(k)} + x_{(k+1)}) & n = 2k(even). \end{cases} \tag{1.7}$$

```
md <- function(x) {
  n <- length(x)
  half <- (n + 1L) %/% 2L
  if (n %% 2L == 1L) {
    sort(x, partial = half)[half]
  } else {
          mean(sort(x, partial = half + 0L:1L)[half + 0L:1L])
        }
}
data <- c(73, 75, 74, 74)
md(data)

[1] 74
```

1.5.3 众数

众数（mode）指在一个样本中出现频率最高的值。例如

```
Grades <- c("A", "D", "C", "D", "C", "C", "C", "C", "F", "B")
names(which.max(table(Grades)))

[1] "C"
```

当样本为连续型数据时，可以用基于分组的数据求众数。例如

```
totalprice <- PASWR::vit2005[["totalprice"]]
DV <- density(totalprice)
yval <- max(DV[["y"]])
ID <- which(DV[["y"]] == yval)
(mode <- DV[["x"]][ID])
```

```
[1] 256944.5
```

1.5.4 分位数

分位数（Quantile）是一组数据分割点。这些分割点能把一个概率分布划分为一系列频率相等的连续区间。一个 q 分位数（q-Quantiles）包含 $q-1$ 个数据分割点，其中每个分割点满足 $0 < k < q$。这些分割点把一个有限集合划分为 q 个元素数量相等的子集。有些 q 分位数有特定的名字，如中位数（median）、四分位数（quartiles）、百分位数（percentiles）等。

一个整体的第 k 个 q 分位数是该分布累积分布函数（cumultative distribution function）穿过 k/q 时的值。所以，一个变量 X 的第 k 个 q 分位数 x 满足以下条件：$\mathbb{P}(X \leq x) \geq k/q$ 和 $\mathbb{P}(X > x) \leq 1 - k/q$。如果用实数 p（$0 < p < 1$）代替 k/q，那么分布的 p 分位数（Quantile）可以用下式表示：

$$Q(p) = F^{-1}(p) = \inf\{x : F(x) \geq k/q\}, \quad 0 < p < 1$$

其中 $F(x)$ 为分布的累计分布函数。

当分布的整体未知时，样本分位数可以被用来对整体分位数进行非参数估计。假设来自整体的独立观测值 $\{X_1, \cdots, X_n\}$ 的样本容量为 n。且该样本的顺序统计量为 $\{X_{(1)}, \cdots, X_{(n)}\}$。一般来讲，在一个容量为 n 的样本中第 k 个顺序统计量所占的分位点可以用下式表示：

$$p_k = \frac{k - \alpha}{n - \alpha - \beta + 1}$$

其中 α 和 β 为两个给定常数。这两个常数决定了样本分位数的具体计算方法。当特定分位点 p_k 对应的 k 不是一个整数时，需要在 $(p_k, X_{(k)})$ 之间线性插入（linear interploation）相应点来求得相应的样本分位数。样本分位数通常通过一个或两个顺序统计量的数值计算而来，其公式如下：

$$\hat{Q}_i(p) = (1 - \gamma)X_{(j)} + \gamma X_{(j+1)}$$

$$\text{其中} \frac{j - m}{n} \leq p \leq \frac{j - m + 1}{n}$$

其中 $m \in \mathbb{R}$，$0 \leq \gamma \leq 1$，且 γ 的值是 $j = \lfloor pn + m \rfloor$ 和 $g = pn + m - j$ 的函数。线性插入使上面两个公式存在如下关系：

$$m = \alpha + p_k(1 - \alpha - \beta)$$
$$\gamma = g$$

R 语言中计算分位数的函数是 quantile()。R 语言提供了 9 种计算分位数的方法，通过参数 type = 1:9 来实现。其中默认的方法第 7 种，$\alpha = \beta = 1$。例如：

```
xx <- c(1, 4, 7, 9, 10, 14, 15, 16, 20, 21)
n <- length(xx)
p <- 0.75
index <- p * (n - 1) + 1
lo <- floor(index)
hi <- ceiling(index)
sorted <- sort(xx, partial = unique(c(lo, hi)))
h <- index - lo
c(index = index, lo = lo, hi = hi, h = h)
sorted[lo] + h * (sorted[hi] - sorted[lo])
(1 - h) * sorted[lo] + h * sorted[hi]
quantile(xx, 0.75)
```

```
index    lo    hi     h
 7.75  7.00  8.00  0.75
[1] 15.75
[1] 15.75
  75%
15.75
```

我们也可以比较一下不同计算方法得出来的不同结果：

```
sapply(1:9,
  function(x) sapply((0:4)*0.25,
    function(y)quantile(xx, y, type = x)))
```

```
     [,1] [,2] [,3] [,4] [,5]  [,6]  [,7]     [,8]    [,9]
0%      1    1    1  1.0    1  1.00  1.00  1.00000  1.0000
25%     7    7    4  5.5    7  6.25  7.50  6.75000  6.8125
50%    10   12   10 10.0   12 12.00 12.00 12.00000 12.0000
75%    16   16   16 15.5   16 17.00 15.75 16.33333 16.2500
100%   21   21   21 21.0   21 21.00 21.00 21.00000 21.0000
```

1.5.5 五数概括法

五数概括法（five number summary）输出的是样本数据中的最小值、下折叶点（lower hinge）、中数、上折叶点（upper hinge）、最大值。其中下折叶点是数据下半部分的中数，上折页点是数据上半部分的中数。具体来说，下折叶点和上折页点分别分别是顺序统计量 $x_{(j)}$ 和 $x_{(n+1-j)}$，其中 j 的计算方式如下：

$$j = \frac{\lfloor \frac{n+1}{2} \rfloor + 1}{2} \tag{1.8}$$

例如：

```
n <- length(xx)
j <- (floor((n + 1) / 2) + 1) / 2
lower.hinge <- sorted[j]
upper.hinge <- sorted[n + 1 - j]
c(min(xx), lower.hinge, median(xx), upper.hinge, max(xx))
```

```
[1]  1  7 12 16 21
```

当然，如果 j 不是整数，需要用线性插入法求相应的值。例如：

```
NYYRBI <- PASWR::Baberuth[7:21, "RBI"]
SNYYRBI <- sort(NYYRBI)
n <- length(SNYYRBI)
j <- (floor((n + 1) / 2) + 1) / 2
c(j, n + 1 - j)
lower.hinge <- SNYYRBI[4] + 0.5 * (SNYYRBI[5] - SNYYRBI[4])
upper.hinge <- SNYYRBI[11] + 0.5 * (SNYYRBI[12] - SNYYRBI[11])
c(min(SNYYRBI), lower.hinge, median(SNYYRBI), upper.hinge, max(SNYYRBI))

[1]  4.5 11.5
[1]  66.0 112.0 137.0 153.5 171.0
```

R 语言中五数概括法是通过函数 fivenum() 来实现的。例如：

```
fivenum(xx)
fivenum(NYYRBI)

[1]  1  7 12 16 21
[1]  66.0 112.0 137.0 153.5 171.0
```

R 语言中的summary() 函数是一个范型函数，即其输出结果会随着输入数据的不同而不同。当输入的数据为数值型数组时，该函数的输出结果将包含：最小值、第一四分位点、第二四分位点（中数）、平均值、第三四分位点、和最大值。例如：

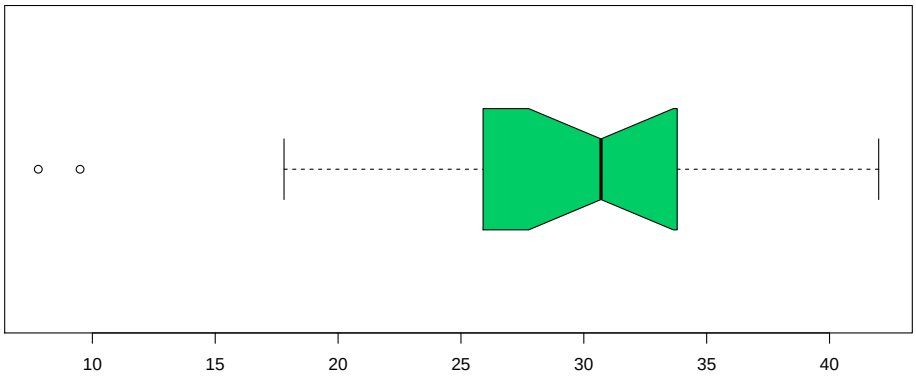
```
summary(xx)
summary(NYYRBI)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
   1.00    7.50   12.00   11.70   15.75   21.00
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   66.0   112.0   137.0   131.3   153.5   171.0
```

1.5.6 箱线图

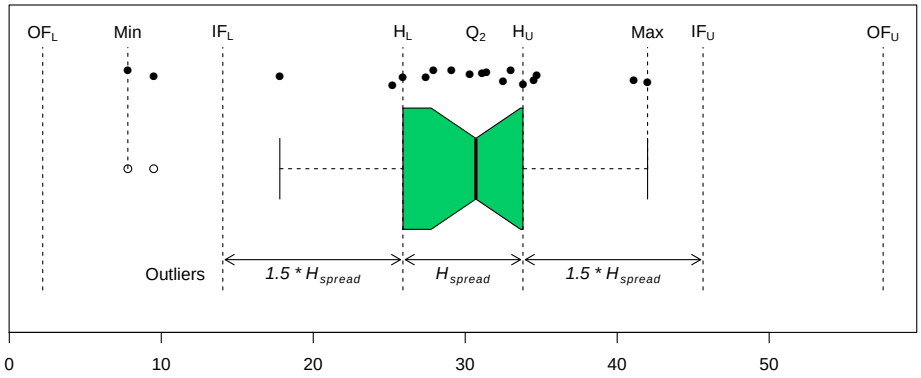
一种展示五数总结的常见方法是箱线图 (boxplots)。R 语言中箱线图是通过函数 boxplot() 来实现的。该函数中还有三个常用参数:horizontal = TRUE 表示把箱线图按水平方式画; notch = TRUE 表示在图中加一个凹口来突出中位数的位置。例如:

```
fat <- PASWR::Bodyfat[["fat"]]
boxplot(fat, col = "springgreen3", horizontal = TRUE, notch = TRUE)
```



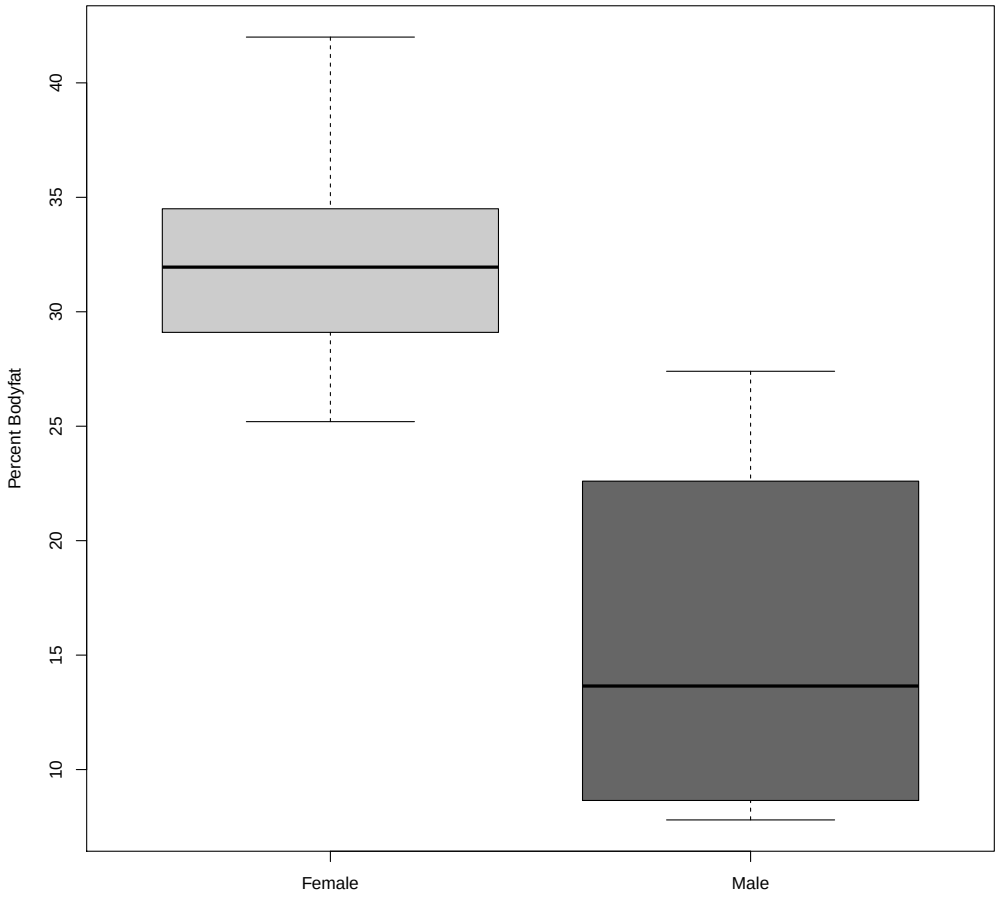
在箱线图中,下折叶(H_L)和上折叶(H_U)之间的距离叫做广度(spread),即 $H_{spread} = H_U - H_L$。比下折叶小 1.5 倍广度的位置叫做下限 (Lower Fence),即 $Fence_L = H_L - 1.5 \times H_{spread}$。比上折叶大 1.5 倍广度的位置叫做上限 (Upper Fence),即 $Fence_U = H_U + 1.5 \times H_{spread}$。所有超出上下限的数据都叫做极端值 (outliers),通常用空心点表示。箱线图中的须状线

(whisker) 标示的是上下折叶和上下临界值 (adjacent value) 的位置。临界值指去除掉临界之后，数据中的最大值和最小值。例如上面的箱线图具有如下含义:



当然箱线图也可以用来描述两个变量之间的关系，例如:

```
bodyfat <- PASWR::Bodyfat
bodyfat[, "sex"] <- factor(bodyfat[, "sex"],
  labels = c("Female", "Male"))
boxplot(fat ~ sex, data = bodyfat,
  col = c("gray80", "gray40"), ylab = "Percent Bodyfat")
```



1.6 离散趋势

1.6.1 全距

描述数据离散性的最简单方法是求全距（Range），即数据中最大值和最小值之间的差值。R 语言中的函数 range(x) 并不直接输出数据的全距，而是输出数据的最小值和最大值。如果要求数据的全距，需要用如下命令

diff(range(x))。例如：

```
x <- 1:10
range(x)
diff(range(x))

[1]  1 10
[1] 9
```

1.6.2 四分位距

描述数据离散性的另一个有效方法是四分位距（interquartile Range, IQR），即顺序数据中处于中间的 50% 的数据之间的距离。其定义为数序统计量中第三个四分位点和第一个四分位点之间的距离，即 $IQR = Q_3 - Q_1$。R 语言中计算四分位距的函数为 IQR(x)。例如：

```
x <- 1:10
quantile(x)
IQR(x)

   0%   25%   50%   75%  100%
 1.00  3.25  5.50  7.75 10.00
[1] 4.5
```

1.6.3 方差和标准差

样本标准差的计算公式如下：

$$s^2 = \sum_{i=1}^{n} \frac{(x - \overline{x})^2}{n - 1} \tag{1.9}$$

R 语言中计算样本标准方差和样本标准差的函数分别是var(x) 和sd(x)。例如：

```
x <- 1:10
var <- sum((x - mean(x)) ^ 2) / (length(x) - 1)
var
sqrt(var)
var(x)
sd(x)

[1] 9.166667
[1] 3.02765
[1] 9.166667
[1] 3.02765
```

1.6.4 离差绝对值中数

离差绝对值中数 (Median absolute deviation，MAD) 是描述数据离散性的另一稳健指标，尤其是当数据呈偏态分布时。其定义如下：

$$MAD = \text{median}|x_i - m|. \tag{1.10}$$

R 语言中计算离差绝对值中数的函数叫做 mad()。当该函数的参数做如下设定时 constant = 1，该函数计算的就是利差绝对值中数。例如：

```
times <- PASWR::SDS4[["Times"]]
median(abs(times - median(times)))
mad(times, constant = 1)

[1] 3
[1] 3
```

1.7 二维数据

1.7.1 二维列联表

R 语言中函数 table() 和 xtabs() 展示类别型数据频率的方式是二维列联表（Two-Way Contingency Tables）。函数 addmargins() 可以在列联表中添加行和列的总结信息。例如：

```
EPIDURAL <- PASWR::EPIDURAL
EPIDURAL[, "Ease"] <- factor(EPIDURAL[, "Ease"],
  levels = c("Easy", "Difficult","Impossible"))
xtab <- xtabs(~ Doctor + Ease, data = EPIDURAL)
addmargins(xtab)
# with(EPIDURAL, addmargins(table(Doctor, Ease)))
```

```
       Ease
Doctor  Easy Difficult Impossible Sum
  Dr. A   19         3          1  23
  Dr. B    7        10          4  21
  Dr. C   18         3          0  21
  Dr. D   13         4          3  20
  Sum     57        20          8  85
```

R 语言中函数 prop.table() 可用来展示相对频率，即比率，而不是绝对频率。函数的参数 margin = ? 用来表示相对频率是基于行（ = 1）还是列（ = 2）来计算的。例如

```
(xtap <- prop.table(xtab, margin = 2))
```

```
       Ease
Doctor       Easy Difficult Impossible
```

```
  Dr. A 0.3333333 0.1500000  0.1250000
  Dr. B 0.1228070 0.5000000  0.5000000
  Dr. C 0.3157895 0.1500000  0.0000000
  Dr. D 0.2280702 0.2000000  0.3750000
```

上述两个函数也可用于展示类别型数据的多维列联表，但展示多维列联表的常用主要函数是 ftable()。例如：

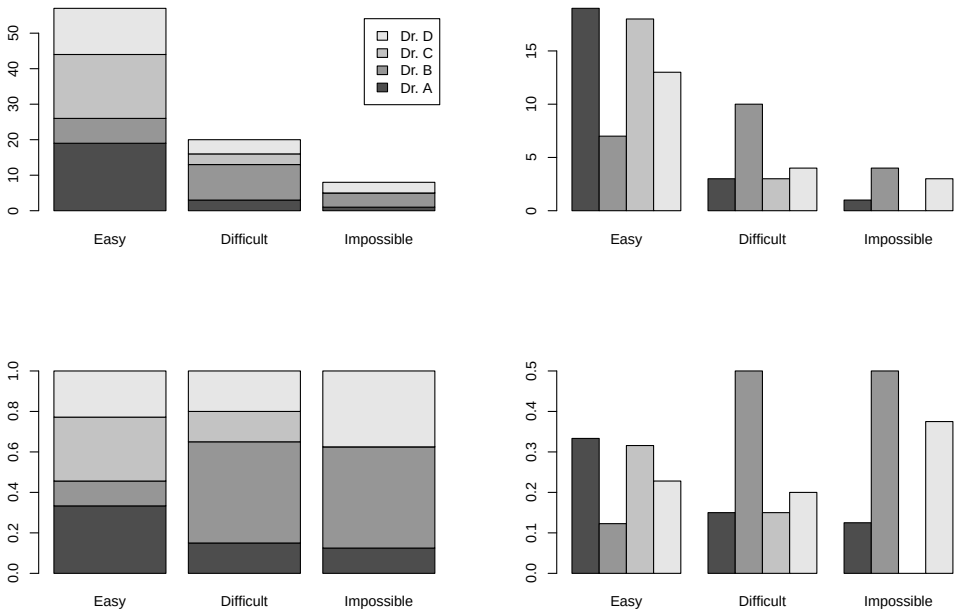
```
with(data = EPIDURAL, ftable(Doctor, Treatment, Ease))
```

```
                          Ease Easy Difficult Impossible
Doctor Treatment
Dr. A  Hamstring Stretch           7         1          0
       Traditional Sitting        12         2          1
Dr. B  Hamstring Stretch           3         3          0
       Traditional Sitting         4         7          4
Dr. C  Hamstring Stretch           8         3          0
       Traditional Sitting        10         0          0
Dr. D  Hamstring Stretch           7         1          2
       Traditional Sitting         6         3          1
```

1.7.2 二维列联表图形

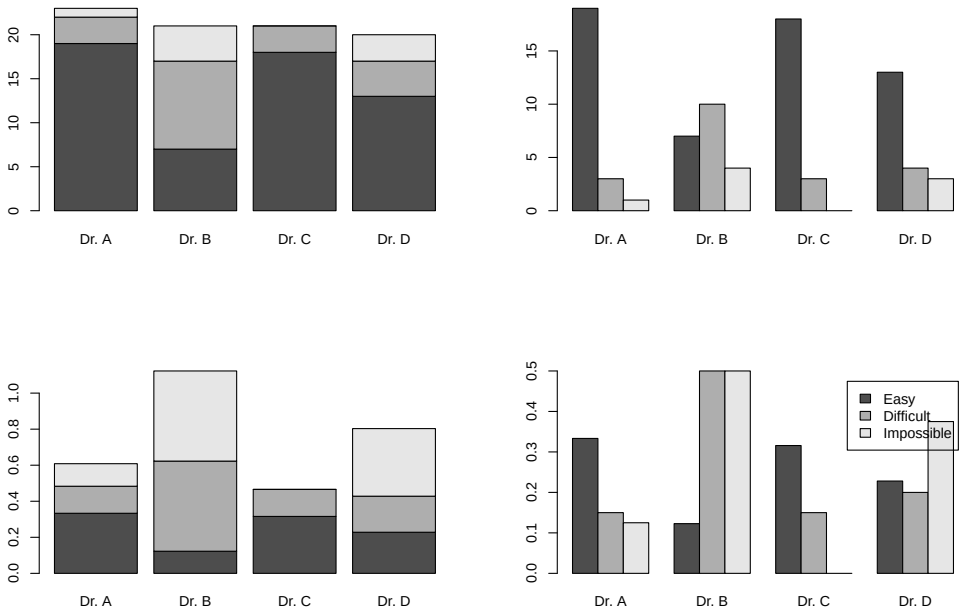
二维列联表数据可以用柱形图来展示。例如：

```
par(mfrow = c(2, 2))
barplot(xtab, legend = TRUE)
barplot(xtab, beside = TRUE)
barplot(xtap)
barplot(xtap, beside = TRUE)
```



我们也可以用函数 t() 对数据进行行列转置。例如:

```
par(mfrow = c(2, 2))
barplot(t(xtab))
barplot(t(xtab), beside = TRUE)
barplot(t(xtap))
barplot(t(xtap), beside = TRUE, legend = TRUE)
```



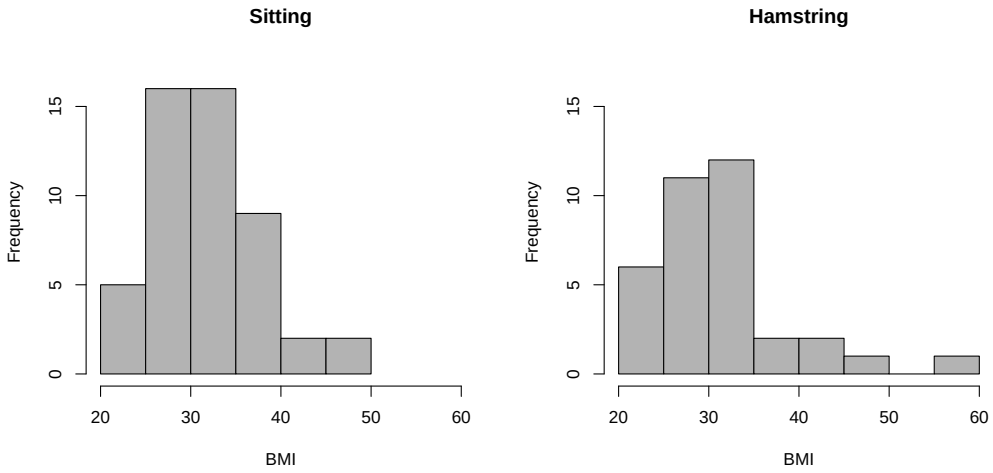
1.7.3 比较不同样本

统计学中通常需要对两个或多个抽样数据进行比较。直方图、概率密度图、箱形图等均可用于比较不同样本的数据。例如假如我们有 BMITS 和 BMIHS 两组数据：

```
EPIDURAL <- PASWR::EPIDURAL
EPIDURAL[, "BMI"] <- EPIDURAL[, "kg"] / (EPIDURAL[, "cm"] / 100) ^ 2
BMITS <- EPIDURAL[EPIDURAL[, "Treatment"] == "Traditional Sitting", "BMI"]
BMIHS <- with(data = EPIDURAL, BMI[Treatment == "Hamstring Stretch"])
```

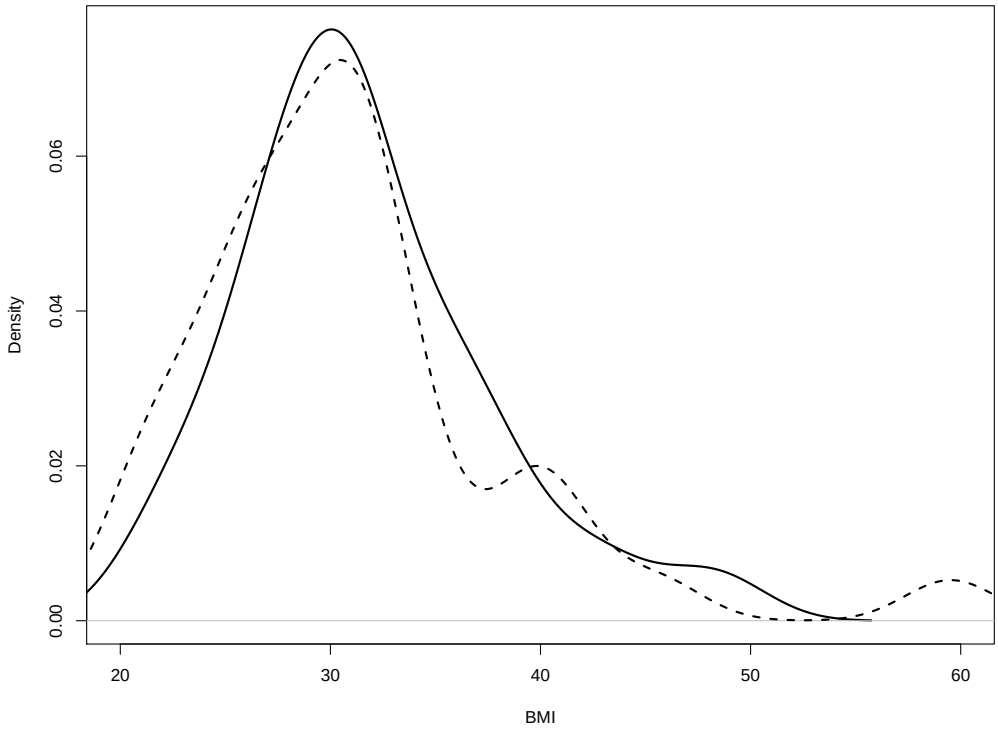
首先，我们可以用直方图来展示这两个样本的异同：

```
par(mfrow = c(1, 2))
hist(BMITS, xlim = c(20, 60), ylim = c(0, 17),
  xlab = "BMI", main = "Sitting", col = 'gray70')
hist(BMIHS, xlim = c(20, 60), ylim = c(0, 17),
  xlab = "BMI", main = "Hamstring", col = 'gray70')
```



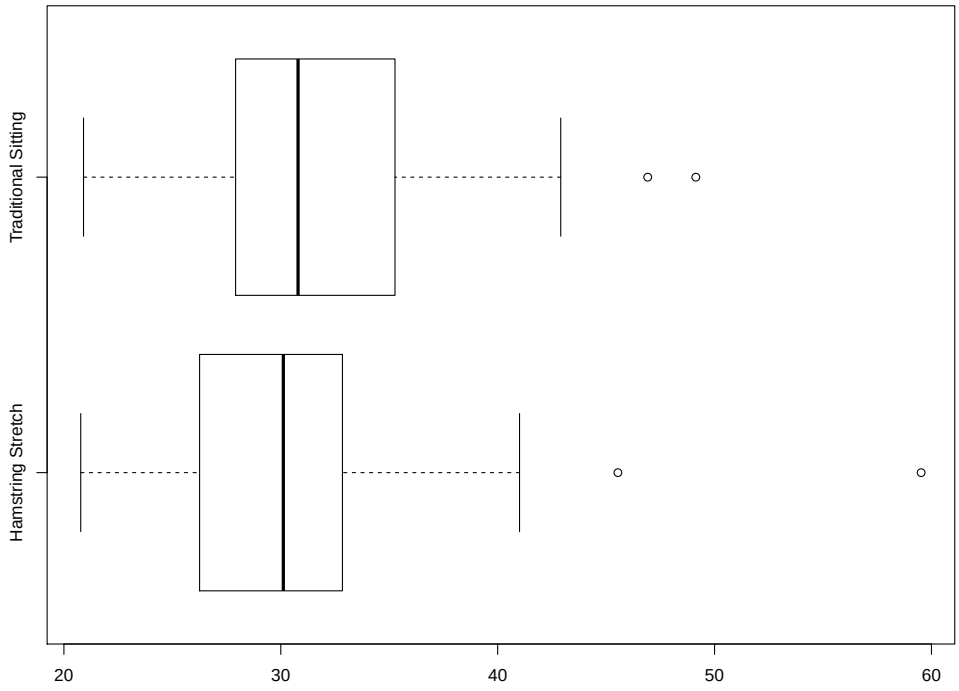
其次，概率密度图也是一种可行方法：

```
plot(density(BMITS), xlim = c(20, 60), lwd = 2, main = "",  xlab = "BMI")
lines(density(BMIHS), lty = 2, lwd = 2)
```



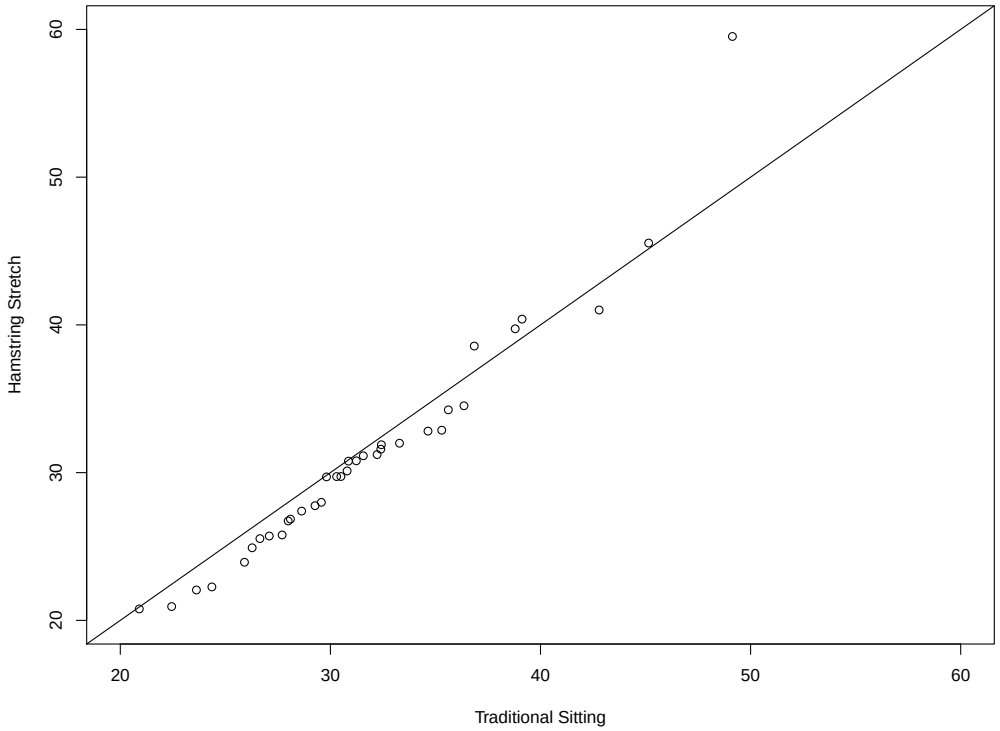
再比如，箱线图也可用来描述这两个样本的差别：

```
boxplot(BMI ~ Treatment, data = EPIDURAL, horizontal = TRUE)
```



另外一种描述两个样本差别的统计图是分位点-分位点图（quantile-quantile plots，Q-Q）。Q-Q 图中，横坐标和纵坐标分别是这两个分布的分位点数。在该图形中，如果两个分布具有相同的形状，则这些点将形成一个直线。R 语言中画 Q-Q 图的函数是 qqplot()。例如

```
qqplot(x = BMITS, y = BMIHS, xlim = c(20, 60),
  ylim = c(20, 60), xlab = "Traditional Sitting",
  ylab = "Hamstring Stretch")
abline(a = 0, b = 1)
```

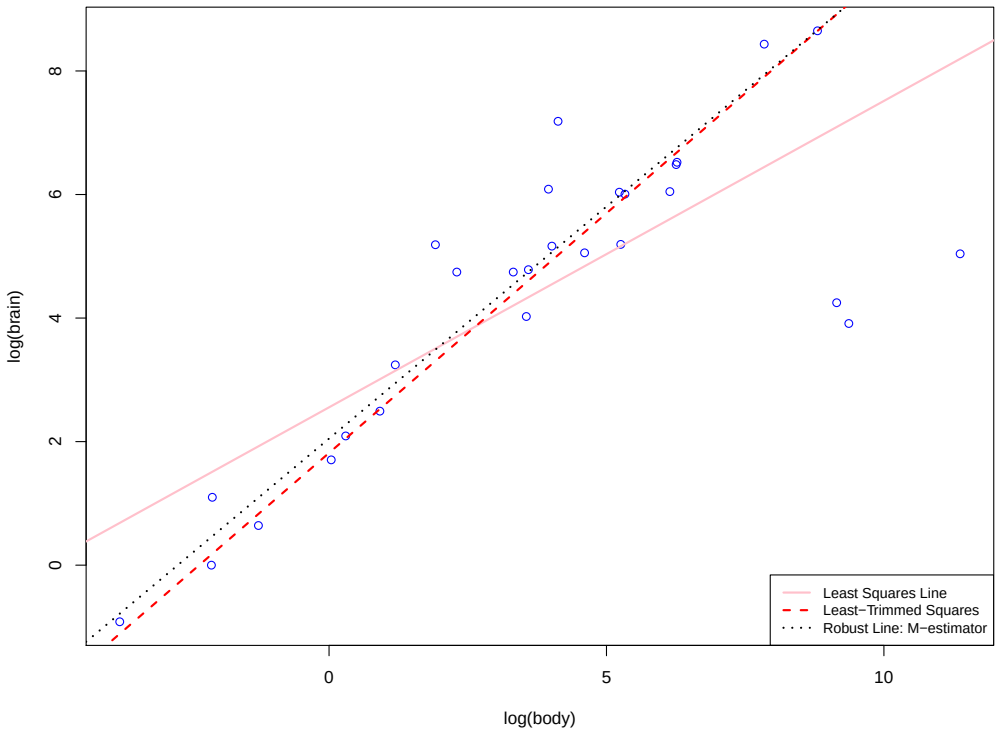


1.7.4 描述变量间关系

描述两个变量关系的最常见图形是散点图（scatterplot）：自变量是横坐标，因变量是纵坐标，每一个数据用一个点表示。散点图中通常会添加一条直线，以描述两个变量之间的整体关系。例如：

```
ff <- with(data = MASS::Animals, log(brain) ~ log(body))
plot(ff, col="blue", xlab = "log(body)", ylab = "log(brain)")
abline(lm(ff), col = "pink", lwd = 2)
abline(lqs(ff), lty = 2, col = "red", lwd = 2)
abline(rlm(ff, method = "MM"), lty = 3, col = "black", lwd = 2)
leglabels <- c("Least Squares Line",
```

```
  "Least-Trimmed Squares", "Robust Line: M-estimator")
colors <- c("pink","red","black")
leglty <- c(1, 2, 3)
legend("bottomright", legend = leglabels, lty = leglty,
  col = colors, lwd = 2, cex = 0.85)
```



1.8 图形组织

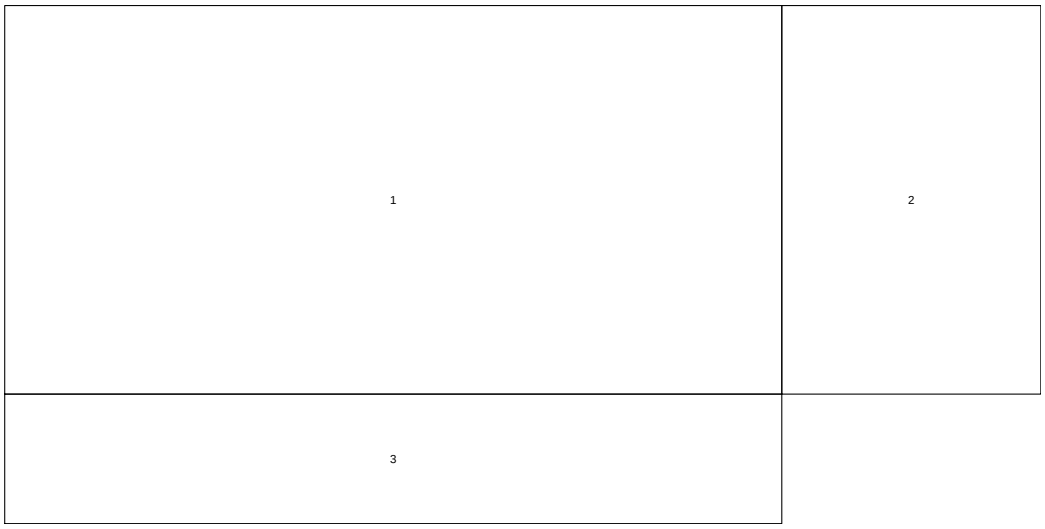
R 语言函数 par(mfrow = c(nr, nc)) 把图形工作区划分为 nr 行和 nc 列个面积相等的区域。而对工作区的更复杂划分可以通过函数 layout() 来实现。假如我们有如下一个矩阵：

```
mat44 <- matrix(c(rep(c(1, 1, 1, 2), 3), rep(3, 3), 0),
  byrow = TRUE, nrow = 4)
mat44
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    2
[2,]    1    1    1    2
[3,]    1    1    1    2
[4,]    3    3    3    0
```

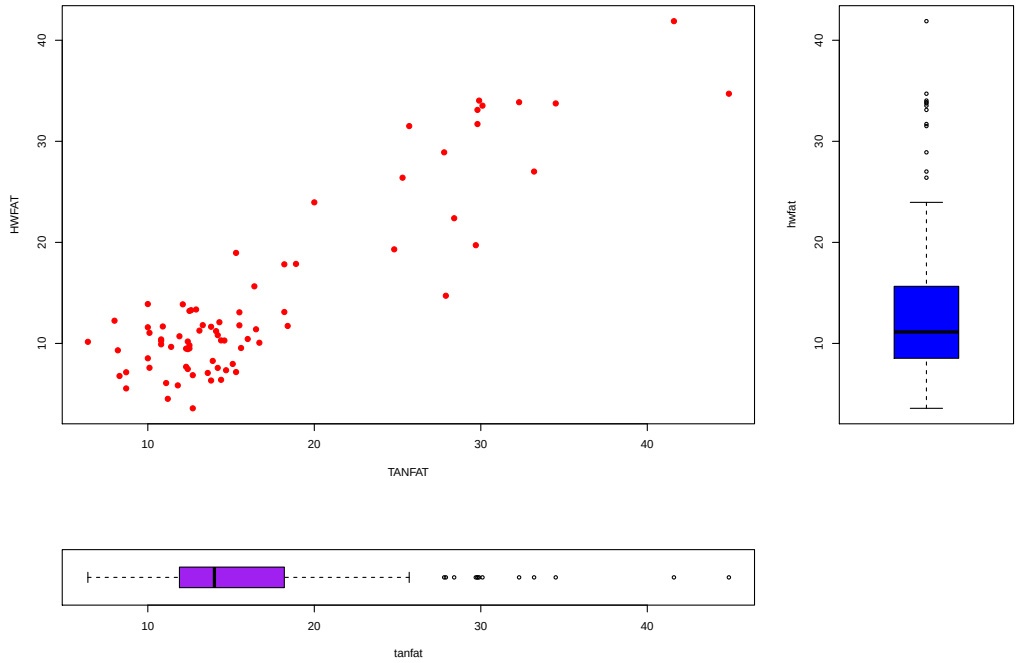
函数 layout() 利用这个矩阵就可以把图形工作区划分为面积不相等的三个区域。然后我们可以利用函数 layout.show() 来展示工作区的实际划分结果。例如

```
layout(mat44)
layout.show(3)
```



划分完的图形工作区包含三个部分，所以我们可以把三个图形分别画在这三个区域中。例如：

```
layout(mat44)
HSwrestler <- PASWR::HSwrestler
plot(HWFAT ~ TANFAT, data = HSwrestler,
  col = "red", pch = 19, main = "")
boxplot(HSwrestler[, "HWFAT"], col = "blue", ylab = "hwfat")
boxplot(HSwrestler[, "TANFAT"], col = "purple",
  horizontal = TRUE, xlab = "tanfat")
```



1.9 多维数据

R 语言中做图的基本函数都如 hist() 和 boxplot() 等都来自于 R 默认加载的基本软件包 graphics。除了这个基本软件包，其他基于 grid 系统的作图软件包包括：vcd、lattice (Sarkar, 2008)、和 ggplot2 (Wickham,

2009)。

1.9.1 类别数据

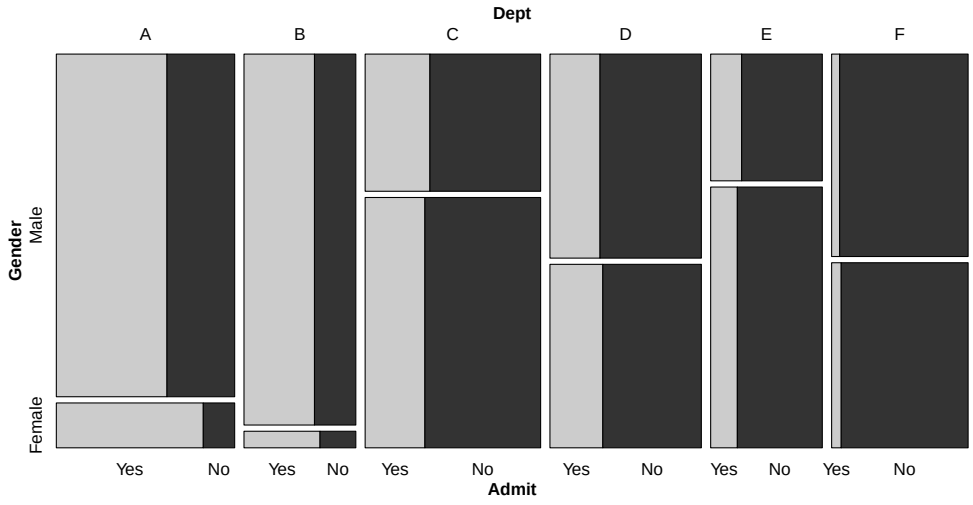
软件包 vcd 中的 mosaic() 函数可用来描述多个类别型变量之间的关系。该函数的一个重要参数叫做表达式,例如 mosaic(formula = ~x1 + x2 + ...)。假定作图工作区是一个边长为 1 的正方形。函数首先根据表达式的第一个自变量 x1 把图形工作区在水平方向上划分成高度不同的模块：模块个数与该自变量的水平数一样；模块高度与该自变量在不同水平上的数据数量成正比。然后，函数根据表达式的第二类别自变量 x2 把每一个水平模块划分为宽度不同的垂直模块：其中垂直模块个数与第二类别自变量的水平数一致，垂直模块的宽度与第二自变量在该水平上的数据数量。以此类推。例如

```
UCB <- datasets::UCBAdmissions
UCB <- as.data.frame(UCBAdmissions)
UCB[, "Admit"] <- factor(UCB[, "Admit"], labels = c("Yes", "No"))
prop.table(xtabs(Freq ~ Dept + Gender + Admit, data = UCB), 1)

, , Admit = Yes

    Gender
Dept       Male     Female
   A 0.54876742 0.09539121
   B 0.60341880 0.02905983
   C 0.13071895 0.22004357
   D 0.17424242 0.16540404
   E 0.09075342 0.16095890
   F 0.03081232 0.03361345

, , Admit = No
```

```
    Gender
Dept       Male     Female
   A 0.33547696 0.02036442
   B 0.35384615 0.01367521
   C 0.22331155 0.42592593
   D 0.35227273 0.30808081
   E 0.23630137 0.51198630
   F 0.49159664 0.44397759
```

上述数据可以用下图表示:

```
vcd::mosaic(~ Dept + Gender + Admit, data = UCB,
  direction = c("v", "h", "v"), highlighting = "Admit",
  highlighting_fill = c("gray80", "gray20"))
```



1.9.2 lattice 作图

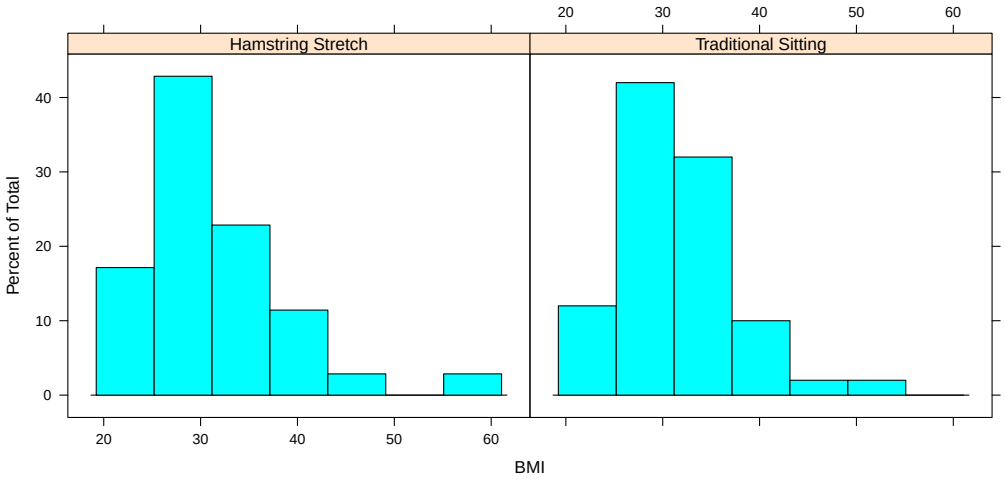
软件包 lattice 是对 Cleveland 于 1993 年创造的特雷里斯图形框架(Trellis displays)在 R 语言中的应用。lattice 包中的常用函数有:barchart()、

bwplot()、densityplot()、dotplot()、histogram()、qq()、qqmath()、stripplot()、xyplot() 等。上述函数均含有一个参数，即句法表达式(formal syntax)。该参数表达了不同变量之间的依存关系，其形式如下：

```
response ~ predictor | conditioning.variable
```

表达式 y ~ x | z 的意思是：在条件变量z 每个给定水平上把变量 y 建模为变量 x(y is modeled as x given z)。条件变量的数据类型通常为因子型（factor），用于把图形划分为不同的面板（panel）。如果条件变量多余一个，可以用乘号把多个条件变量连起来，如 y ~ x | z1 * z1。例如

```
EPIDURAL <- PASWR::EPIDURAL
EPIDURAL[, "BMI"] <- EPIDURAL[, "kg"] / (EPIDURAL[, "cm"] / 100) ^ 2
fs <- ~ BMI | Treatment
lattice::histogram(fs, data = EPIDURAL, layout = c(2, 1))
```



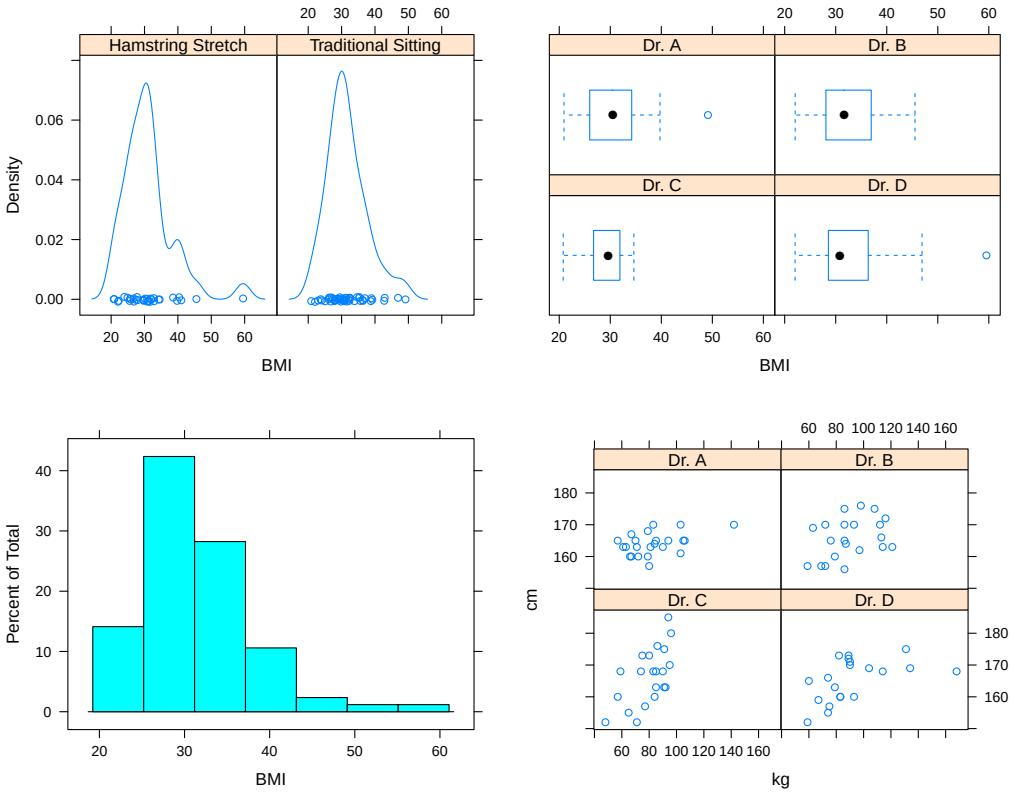
1.9.3 多个图形

如果需要在同一个图形工作区中添加多个图形，lattice 有自己的句法结构：

```
print(latticegraph,
  split = c(column, row, number_of_columns, number_of_rows),
  more = TRUE/FALSE)
```

该函数首先把图形工作区切分为 number_of_columns 列和 number_of_rows 行，并把名称为 latticegraph 的图形添加到位于第 column 列、第 row 行的单元格上。参数 more = 都设置为 TRUE 除非要添加的图形是最后一个。例如：

```
graph1 <- lattice::histogram(~ BMI, data = EPIDURAL)
graph2 <- lattice::xyplot(
  cm ~ kg | Doctor, data = EPIDURAL, as.table = TRUE)
graph3 <- lattice::densityplot(
  ~ BMI | Treatment, data = EPIDURAL, as.table = TRUE)
graph4 <- lattice::bwplot(
  ~ BMI | Doctor, data = EPIDURAL, as.table = TRUE)
print(graph1, split = c(1, 2, 2, 2), more = TRUE)
print(graph2, split = c(2, 2, 2, 2), more = TRUE)
print(graph3, split = c(1, 1, 2, 2), more = TRUE)
print(graph4, split = c(2, 1, 2, 2), more = FALSE)
```



如前所述,lattice 的图形工作区形成了一个以左下角为坐标原点(0,0)、右上角为坐标终点(1,1)的面积为 1 的坐标系。所以 lattice 还可以通过坐标位置确定图形的摆放位置。其句法结构如下:

```
print(latticegraph,
  position = c(x_LL, y_LL, x_UR, y_UR), more = TRUE/FALSE)
```

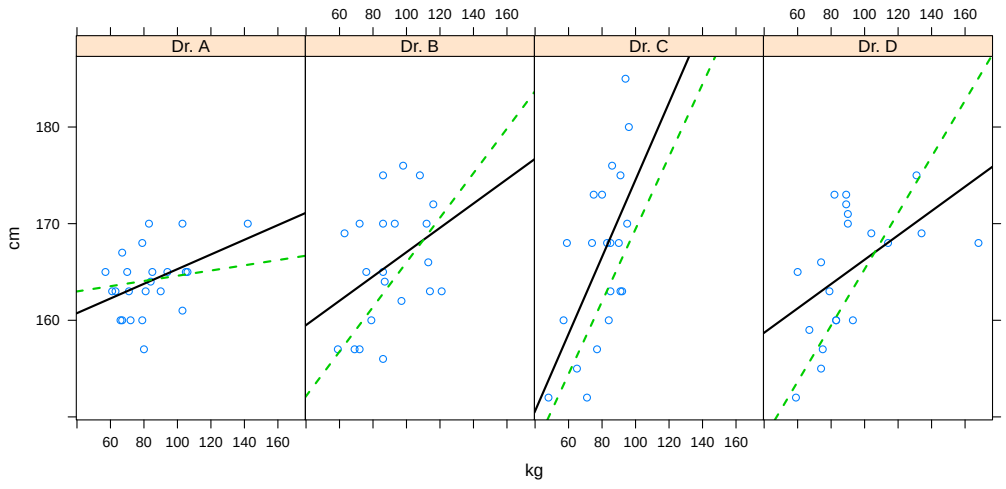
该函数说的是把名称为 latticegraph 的图形添加到左下角为(x_LL, y_LL)、右上角为(x_UR, y_UR)的一个矩形区域中。例如下面的命令将与上面的命令产生类似的结果:

```
print(graph1, position = c(0, 0, 0.5, 0.5), more = TRUE)
print(graph2, position = c(0.5, 0, 1, 0.5), more = TRUE)
print(graph3, position = c(0, 0.5, 0.5, 1), more = TRUE)
print(graph4, position = c(0.5, 0.5, 1, 1), more = FALSE)
```

1.9.4 面板内容

lattice 包还允许对每个面板（panel）进行控制添加一些特征。例如函数 panel.abline() 会在面板中添加一个回归线。lattice 包允许的所有面板控制函数可以通过 ?panel.functions 来查看。例如：

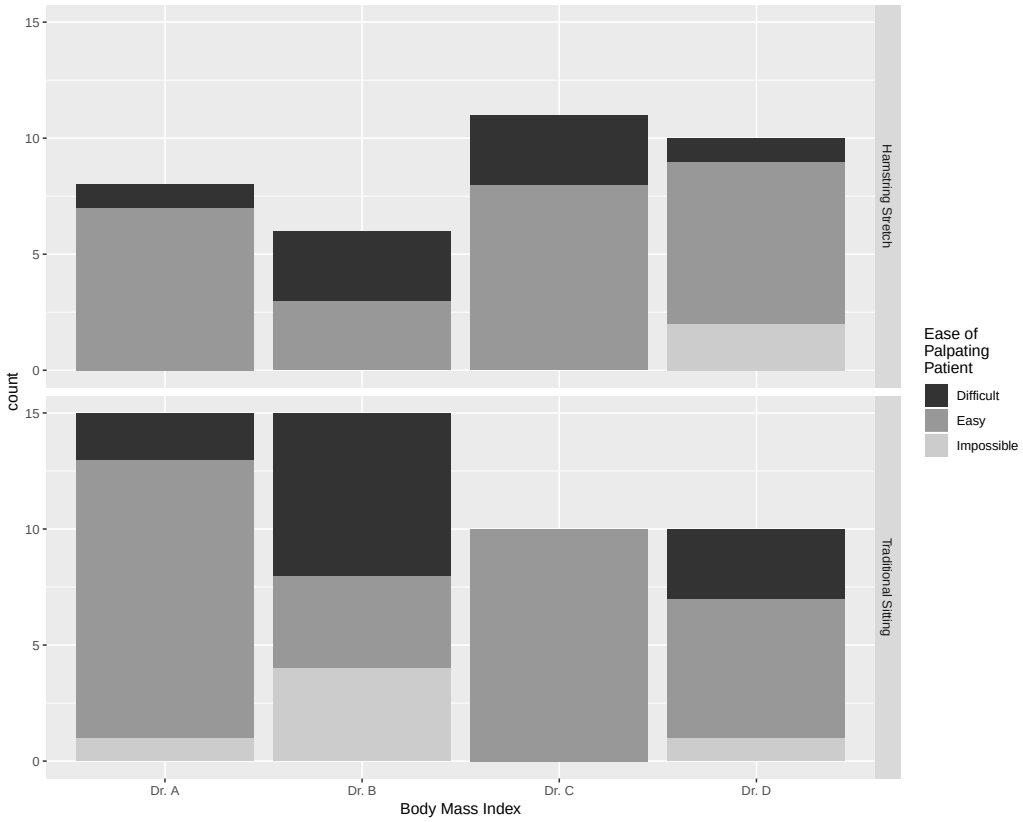
```
panel.scatreg <- function(x, y) {
  lattice::panel.xyplot(x, y)
  lattice::panel.abline(lm(y ~ x), lwd = 2)
  lattice::panel.abline(MASS::lqs(y ~ x), col = 3, lty = 2, lwd = 2)
}
lattice::xyplot(cm ~ kg | Doctor, data = EPIDURAL,
  as.table = TRUE, panel = panel.scatreg)
```



1.9.5 ggplot2 作图

R 包 ggplot2 是按层画图的，它有自己独特的语法结构。首先输入的数据必须是一个数据框（data frame）。其次一个 ggplot2 图形首先把存储为数据框的数据通过美学函数（aesthetic function）匹配到一个几何对象（geometric object）中。美学函数 aes() 描述了数据框中的变量如何匹配到图形的不同视觉特征上（美学特征）。而几何对象函数 geom 决定了如何展示数据，散点图还是箱形图?。下面是一个简单的例子:

```
library(ggplot2)
p <- ggplot(data = EPIDURAL, aes(x = Doctor, fill = Ease))
p <- p + geom_bar(position = "stack")
p <- p + facet_grid(Treatment ~ .)
p <- p + labs(x = "Body Mass Index")
p <- p + guides(fill = guide_legend("Ease of\nPalpating\nPatient"))
p <- p + scale_fill_grey()
p
```


15
10
5
0
count
Hamstring Stretch
Traditional Sitting
Dr. A
Dr. B
Dr. C
Dr. D
Body Mass Index
Ease of Palpating Patient
Difficult
Easy
Impossible

第二章　概率和随机变量

2.1　简介

2.2　数数技术

2.2.1　排列

从 n 个相异元素中取出 k 个元素,如果 k 个元素可以重复出现(sampling with replacement)，则 k 个元素的排列数量为 n^k。如果 k 个元素不能重复出现（sampling without replacement)，则 k 个元素的排列（permutation）数量为：

$$P_{k,n} = n(n-1)(n-2)\cdots(n-k+1) = \frac{n!}{(n-k)!}$$

其中 $n!$ 表示 n 的阶乘。R 语言中计算阶乘的函数是 `factorial()`。例如，从 4 个顺序元素中选取 3 个的排列数为：

```
factorial(4) / factorial(4 - 3)
```

```
[1] 24
```

另外，当 n 个数中，x_1 重复了 n_1 次，x_2 重复了 n_2 次，x_k 重复了 n_k 次，那么该 n 个元素有下面几种可能的排列方式：

$$\frac{n!}{n_1! \cdot n_2! \cdots n_k!}$$

例如，利用 DATA 这四个字母我们可以形成如下中排列方式：

```
factorial(4)/(factorial(2) * factorial(1) * factorial(1))
```

```
[1] 12
```

2.2.2 组合

从 n 个元素中取出 k 个元素，如果不考虑顺序，那么 k 个元素的组合（combination）数量为：

$$C_{k,n} = \left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$$

R 语言中计算组合数量的函数为 `choose(n, k)`。例如：从 8 个人中选取 3 个人共有如下一些可能性：

```
choose(n = 8, k = 3)
```

```
[1] 56
```

2.3 概率公理

2.3.1 样本空间和事件

试验（experiment）是为了查看某事的结果或某物的性能而从事的某种活动。满足以下三个特点的试验被称为随机试验（random experiment）：

1. 可重复性：在相同条件下可重复进行；

2. 可观测性：每次试验的可能结果不止一个，并且能实现明确试验的所有可能结果；

3. 不确定性：一次试验之前，不能预知会出现哪一个结果。

随机试验 E 的所有可能结果组成的集合称为 E 的样本空间（sample space），计为 Ω。样本空间中的元素，即试验 E 的每个结果，被称为样本点（sample point）。试验 E 的样本空间的子集称为 E 的随机事件（random event），简称事件（event）。特别，有一个样本点组成的单点集，称为基本事件（simple event）。其他事件称为复合事件（conpound event）。在每次试验中，当且仅当这一子集中的一个样本点出现时，称这一事件发生（event occurs）。

2.3.2 集合理论

事件是一个集合，所以事件间的关系和事件的运算自然按照集合论中集合之间的关系和集合运算来处理。

2.3.3 什么是概率

在相同条件下进行了 n 次试验。在这 n 次试验中，事件 E 发生的次数 n_E 称为事件 E 发生的频数。比值 n_E/n 称为事件 E 发生的相对频率（relative frequncy），并记做 $f_n(E)$。若随着试验次数 n 的增加，相对频率稳定在常数 p 附近，则称常数 p 为事件 E 发生的概率（probability），即

$$\mathbb{P}(E) = \lim_{n\to+\infty} \frac{n_E}{n} = p$$

概率的上述定义必须给予如下假设：随着试验次数的增加，事件发生的相对频率会汇聚（converge）到某一个特定值附近。这是一个复杂的前提假设，二下述基于三个公理的概率定义则不需要这个假设。

设随机试验的样本空间为 Ω。若对于每一事件 A，有且仅有一个实数 $P(A)$ 与之对应。若 $P(A)$ 满足以下公理，则称 $P(E)$ 称为事件 E 的概率：

1. 非负性：$0 \leq \mathbb{P}(A) \leq 1$

2. 规范性：$\mathbb{P}(\Omega) = 1$

3. 可列可加性：若可列事件 $A_1, A_2, \cdots$ 两两相斥，则

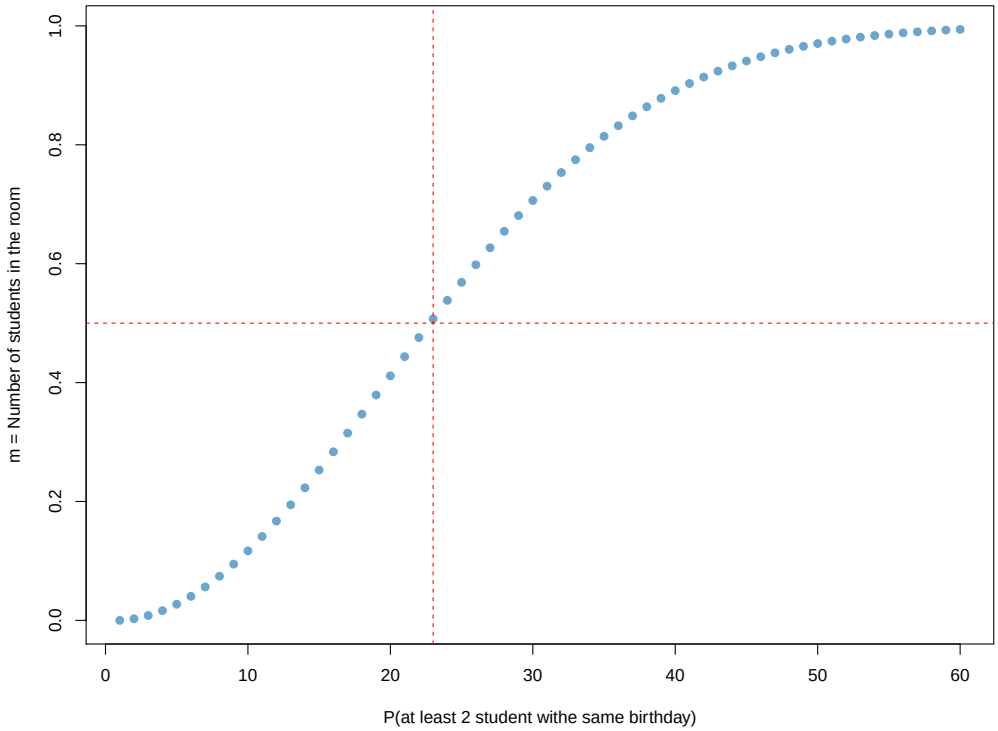
$$\mathbb{P}(\bigcup_{i=1}^{+\infty} A_i) = \sum_{i=1}^{+\infty} \mathbb{P}(A_i)$$

根据上述公理，概率具有以下特征：

$$\begin{aligned} &\mathbb{P}(E^c) = 1 - \mathbb{P} \\ &\mathbb{P}(E \bigcup F) = \mathbb{P}(E) + \mathbb{P}(F) - \mathbb{P}(E \bigcap F) \\ &\mathbb{P}(\varnothing) = 0 \\ &\mathbb{P}(E) \leq \mathbb{P}(F), \text{若 } E \subset F \end{aligned} \tag{2.1}$$

例如：有一个人数为 m 的班级，请问班级中至少又两个同学在同一天出生的概率是多少？

```
m <- 1:60
ff <- function(i) 1 - prod((365:(365 - i + 1)) / 365)
ProbAtL2SB <- sapply(m, ff)
plot(m, ProbAtL2SB, col = "skyblue3", pch = 19,
  xlab = "P(at least 2 student withe same birthday)",
  ylab = "m = Number of students in the room")
abline(h = 0.5, lty = 2, col = "red")
abline(v = 23, lty = 2, col = "red")
```



2.3.4 条件概率

设 A，B 是两个事件，且 $P(A) > 0$，则称 $P(B|A)$ 为在事件 A 发生的条件下事件 B 发生的条件概率（Conditional probability）：

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A \bigcap B)}{\mathbb{P}(A)}$$

例如，抵两次骰子，在第一次为 4 的情况下，两次点数和为 8 的概率是多少？

```
Omega <- expand.grid(roll1 = 1:6, roll2 = 1:6)
G <- subset(Omega, roll1 == 4)
HaG <- subset(Omega, roll1 == 4 & (roll1 + roll2) == 8)
PG <- nrow(G) / nrow(Omega)
PHaG <- nrow(HaG) / nrow(Omega)
PHgG <- PHaG / PG
MASS::fractions(PHgG)

[1] 1/6
```

2.3.5 贝叶斯定理

全概率公式（Law of Total Probability）：设 Ω 为试验 E 的样本空间，$B_1, B_2, \cdots, B_n$ 为 E 的一组事件。若 $\bigcup_{i=1}^{n} B_i = \Omega$，$B_i \bigcap B_j = \varnothing, i \neq j$，且 $\mathbb{P}(B_i) > 0$。对于样本空间中任何事件 A 有：

$$\mathbb{P}(A) = \sum_{i=1}^{n} \mathbb{P}(A \bigcap B_i) = \sum_{i=1}^{n} \mathbb{P}(A|B_i)\mathbb{P}(B_i).$$

把全概率公式代入条件概率公式可得贝叶斯公式（Bayes' Rule）：

$$\mathbb{P}(B_i|A) = \frac{\mathbb{P}(A \bigcap B_i)}{\mathbb{P}(A)} = \frac{\mathbb{P}(A|B_j)\mathbb{P}(B_j)}{\sum_{i=1}^{n} \mathbb{P}(A|B_i)\mathbb{P}(B_i)}$$

例如，三门问题（Monty Hall problem）。假设你正在参加一个游戏节目，你被要求在三扇门中选择一扇：其中一扇后面有一辆车；其余两扇后面则是山羊。你选择了一道门，假设是一号门，然后知道门后面有什么的主持人，开启了另一扇后面有山羊的门，假设是三号门。他然后问你："你想选择二号门吗？"转换你的选择对你来说是一种优势吗？

令 D_i 表示 i 号门后面有汽车；O_j 表示主持人打开的是 j 号门。假如主持人打开的是三号门，那么不改变选项而赢得汽车的概率是 $\mathbb{P}(D_1|O_3)$，而

改变选项而赢得汽车的概率为 $\mathbb{P}(D_2|O_3)$。根据贝叶斯公式得知：

$$\mathbb{P}(D_1|O_3) = \frac{\mathbb{P}(O_3|D_1)\mathbb{P}(D_1)}{\sum_{i=1}^{n}\mathbb{P}(O_3|D_i)\mathbb{P}(D_i)}$$
$$\mathbb{P}(D_2|O_3) = \frac{\mathbb{P}(O_3|D_2)\mathbb{P}(D_2)}{\sum_{i=1}^{n}\mathbb{P}(O_3|D_i)\mathbb{P}(D_i)}$$

开始时选手打开任意一扇门的概率是 $\mathbb{P}(D_1) = \mathbb{P}(D_2) = \mathbb{P}(D_3) = 1/3$。如果三号门后面是汽车，主持人可以打开二号或三号门，所以打开三号门的概率为 1/2，即 $\mathbb{P}(O_3|D_1) = 1/2$。如果二号门后面是汽车，主持人只能打开三号门，所以打开三号门的概率为 1，即 $\mathbb{P}(O_3|D_2) = 1$。如果三号门后是汽车，则主持人不能打开三号门，所以此时打开三号门的概率为 0，即 $\mathbb{P}(O_3|D_3) = 0$。把上述值代入贝叶斯公式，得出改变选择后得到汽车的概率比不改变获得汽车的概率高一倍。用 R 语言模拟该数据可以得出类似的结果：

```
set.seed(2)
actual <- sample(x = 1:3, size = 10000, replace = TRUE)
aguess <- sample(x = 1:3, size = 10000, replace = TRUE)
equals <- (actual == aguess)
PNoSwitch <- sum(equals) / 10000
not.eq <- (actual != aguess)
PSwitch <- sum(not.eq) / 10000
Probs <- c(PNoSwitch, PSwitch)
names(Probs) <- c("P(Win no Switch)", "P(Win Switch)")
Probs

P(Win no Switch)    P(Win Switch)
          0.3348           0.6652
```

另外，满足下面等式的两个事件 A、B 相互独立（independent）：

$$\mathbb{P}(A\bigcap B) = \mathbb{P}(A)\mathbb{P}(B)$$

2.4 随机变量

设随机试验 E 的样本空间为 $\Omega = \{\omega\}$。$X = X(\omega)(\omega \in \Omega)$ 是定义在样本空间上的实值单值函数，称 $X = X(e)$ 为随机变量 (Random variable)。

设 X 是一个随机变量，x 是任意实数，函数:

$$F(x) = \mathbb{P}\{X \leq x\}, -\infty < x < \infty$$

称 X 的累积分布函数（cumulative distribution function, cdf）或简称分布函数。分布函数具有如下特征:

1. $F(x)$ 是一个不减函数，即若 $a < b$ 则 $F(a) \leq F(b)$。

2. $0 \leq F(x) \leq 1$，且 $F(-\infty) = \lim\limits_{x \to -\infty} = 0$; $F(\infty) = \lim\limits_{x \to +\infty} = 1$。

3. 右连续，即 $F(x + 0) = F(x)$。

2.4.1 离散型随机变量

有些随机变量全部可能取到的值是有限个或可列无限多个，称离散型随机变量（discrete variable)。离散型随机变量的累积分布函数（cdf）可定义为:

$$F(x) = \mathbb{P}(X \leq x) = \sum_{k \leq x} p(k)$$

假设一个离散型随机变量的数据点个数为 n，累积分布函数的估计值，即经验性累积分布函数（empirical cumulative distribution function, ecdf）可写成:

$$\hat{F}_n(t) = \sum_{i=1}^{n} \frac{I\{x_i \leq t\}}{n}.$$

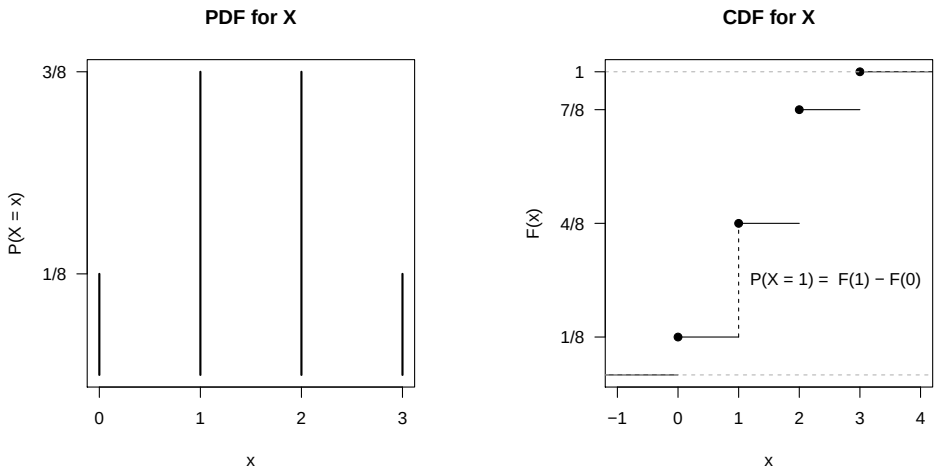
离散变量的概率质量函数(probability mass function，pmf)或概率密度函数（probability density function，pdf)：

$$P\{X = x_k\} = p_k \quad (k = 1\,, 2\,, \cdots)$$

概率质量函数的估计值，即经验概率质量函数（empirical probability mass function，epmf）或经验概率密度函数（empirical probability density function, epdf）可写成：

$$\hat{f}_n(t) = \sum_{i=1}^{n} \frac{I\{x_i = t\}}{n}.$$

R 语言中函数 `ecdf()` 是用来计算经验性累计分布函数的。例如：



2.4.2 连续型随机变量

如果对于随机变量 X 的分布函数 $F(x)$，存在非负函数 $f(x)$，使得对于任意实数 x 有

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

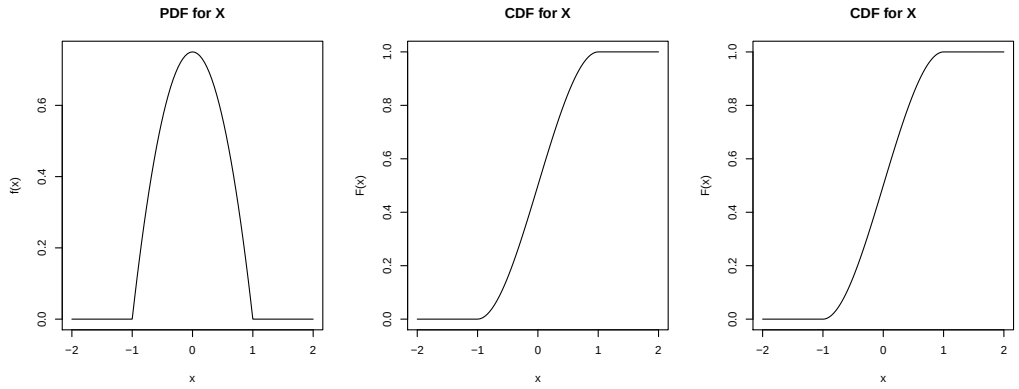
则称 X 为连续型随机变量（Continuous Random Variable），其中 $f(x)$ 称为 X 的概率密度函数（Probability Density Functions），简称概率密度。概率密度 $f(x)$ 具有以下性质：

1. $f(x) \geq 0$, $-\infty < x < +\infty$;
2. $\int_{-\infty}^{\infty} f(x)dx = 1$;
3. $\mathbb{P}(a \leq X \leq b) = \int_a^b f(x)dx$
4. 若 $f(x)$ 在点 x 处连续，则有 $F'(x) = f(x)$。

R 语言中函数 integrate() 可以用来计算函数在某个变量的有限和无限区间上的积分。例如：

```
ff <- function(x){
  y <- 3 / 4 * (1 - x ^ 2)
  y[x < -1 | x > 1] <- 0
  return(y)
}
FF <- function(x){
  y <- -x ^ 3 / 4 + 3 * x / 4 + 1 / 2
  y[x <= -1] <- 0
  y[x > 1] <- 1
  return(y)
}
Ff <- function(x) integrate(ff, -Inf, x)["value"]
xx <- seq(-2, 2, by = 0.01)
yy <- sapply(xx, Ff)
par(mfrow = c(1, 3))
curve(ff, -2, 2, xlab = "x", ylab = "f(x)", main = "PDF for X")
curve(FF, -2, 2, xlab = "x", ylab = "F(x)", main = "CDF for X")
```

```
plot(xx, yy, type = "l", xlab = "x", ylab = "F(x)",
  main = "CDF for X")
```




```
library(ggplot2)
p <- ggplot(data.frame(x = c(-2, 2)), aes(x = x))
p + stat_function(fun = ff) + labs(x = "x", y = "f(x)",
  title = "PDF for X")
p + stat_function(fun = FF) + labs(x = "x", y = "F(x)",
  title = "CDF for X")
```

2.5 随机变量的数学特征

2.5.1 众数、中数和百分位数

众数（mode）在离散型随机变量中指概率分布中最经常出现的 x 值；在连续型随机变量中指使得概率密度函数最大化的 x 值。

中数（median）指满足以下条件的 m 值：

$$m = \begin{cases} \mathbb{P}(X \le m) \ge 1/2, \text{且} \quad \mathbb{P}(X \ge m) \ge 1/2 & \text{离散变量} \\ \int_{-\infty}^{m} f(x)dx = \int_{m}^{\infty} f(x)dx = \frac{1}{2} & \text{连续变量} \end{cases}$$

第 j^{th} 个百分位点（percentile）指满足下面公式的 x_j 值：

$$x_j = \begin{cases} \mathbb{P}(X \leq x_j) \geq j/100\,, \quad \text{且}\mathbb{P}(X \geq x_j) \geq j/100 & \text{离散变量} \\ \int_{-\infty}^{x_j} f(x)dx = \frac{j}{100} & \text{连续变量} \end{cases}$$

2.5.2 期望值

设一个随机变量 X 的概率密度函数为 $p(x)$（离散变量）或 $f(x)$（连续变量），则随机变量 X 的期望值（expected value）为：

$$E[X] = \mu_X = \begin{cases} \sum\limits_{x} x \cdot p(x) & \text{离散变量} \\ \int_{-\infty}^{\infty} x \cdot f(x)dx & \text{连续变量} \end{cases}$$

若 Y 是随机变量的函数，即 $Y = g(X)$，那么该函数的期望值为

$$E[Y] = E[g(X)] = \begin{cases} \sum\limits_{x} g(x) \cdot p(x) & \text{离散变量} \\ \int_{-\infty}^{\infty} g(x) \cdot f(x)dx & \text{连续变量} \end{cases}$$

期望值具有如下一些特点：$E[bX] = bE[X]$，$E[a + bX] = a + bE[X]$ 。

例如，在一个游戏的圆盘上有 1、5、30 三个数，圆盘落在这三个数上的概率分别为 0.50、0.45 和 0.05。游戏参与者每转一次圆盘需花 5 块钱，并能获得圆盘停止位置的钱数，问该游戏公平吗（期望值与花的钱数一样）?

```
x <- c(1, 5, 30)
px <- c(0.50, 0.45, 0.05)
EX <- sum(x * px)
WM <- weighted.mean(x, px)
c(EX, WM)

[1] 4.25 4.25
```

2.5.3 动差和动差生成函数

定义在实数域上的实函数相对于常数 c 的 n 阶动差（moment，或矩）为

$$\mu_n = E[(X-k)^k] = \begin{cases} \sum_k (x-c)^k \cdot P(x) & \text{离散变量} \\ \int_{-\infty}^{\infty} (x-c)^n f(x)dx & \text{连续变量} \end{cases}$$

如果 $f(x)$ 是概率密度函数，函数的期望值就是概率密度函数的一阶原点矩；而函数的方差（variance）、偏态（skewness）、峰态（Kurtosis）分别是概率密度函数的二阶、三阶和四阶中心矩。

一个随机变量的方差（variance）为

$$Var[X] = \sigma_X^2 = E[(X-\mu)^2] = \begin{cases} E[X^2] - \mu^2 & \text{离散变量} \\ \displaystyle\int_{-\infty}^{\infty} (x-\mu)^2 \cdot f(x)dx & \text{连续型量} \end{cases}$$

设一个随机变量 X 的概率密度函数为 $f(x)$（连续变量）或 $p(x)$（离散变量），则该变量的动差生成函数（Moment generating function，mgf）为：

$$M_X(t) = E[e^{tX}] = \begin{cases} \sum_x e^{tx} p(x) & \text{离散变量} \\ \int_{-\infty}^{\infty} e^{tx} f(x)dx\,, \quad -h < t < h & \text{连续变量} \end{cases} \tag{2.2}$$

2.5.4 切比雪夫不等式和大数定律

设 $g(X)$ 是定义在随机变量 X 上的函数且 $g(X) \geq 0$，那么对于任何正数 K 来说，下面不等式均成立：

$$\mathbb{P}(g(X) \geq K) \leq \frac{E[g(X)]}{K}$$

设 $g(X) = (X-\mu)^2, K = k^2\sigma^2$，计算可得切比雪夫不等式（Chebyshev's Inequality）：

$$\mathbb{P}(|X-\mu| \geq k) \leq \frac{\sigma^2}{k^2}$$

切比雪夫不等式（Chebyshev's Inequality）的一个重要应用是弱大数定律（weak law of large numbers）：设 $X_1, X_2, \cdots, X_n$ 是相互独立，服从统一分布的随机变量序列，具有数学期望 $E(X_k) = \mu \quad (k = 1, 2, \cdots)$。作前 n 个变量的算数平均 $\frac{1}{n}\sum_{i=1}^{n} X_k$，则对于任意 $\varepsilon > 0$，有

$$\lim_{n\to\infty} \mathbb{P}\left\{\left|\frac{1}{n}\sum_{k=1}^{n} X_k - \mu\right| \geq \varepsilon\right\} = 0 \tag{2.3}$$

第三章　单变量概率分布

3.1　简介

3.2　离散型单变量

3.2.1　离散型均匀分布

对于一个样本量为 n 的随机变量 X 而言，如果变量取得任何一个样本点的概率均相等，则称随机变量服从参数值为 n 的离散型均匀分布（discrete uniform distribution）：

$$\boxed{\begin{aligned}\mathbb{P}(X=x_i|n)&=\frac{1}{n},\quad i=1,2,\cdots,n.\\ E[X]&=\frac{1}{n}\sum_{i=1}^{n}x_i\\ Var[X]&=\frac{1}{n}\sum_{i=1}^{n}(x_i-E[X])^2\\ M_x(t)&=\frac{1}{n}\sum_{i=1}^{n}e^{tx_i}\end{aligned}} \tag{3.1}$$

3.2.2　二项式分布

设随机试验 E 只有两种可能结果：事件 A 发生或事件 A 不发生，则称 E 为伯努利试验（Bernoulli trial）。记 $P(A)=\pi\quad(0<\pi<1)$，此时

$P(\overline{A}) = 1 - \pi = \varrho$。若把事件 A 发生记做 1，事件 A 不发生记做 0，则此随机变量服从参数为 π 的伯努利分布或 $0-1$ 分布：

$$
\begin{array}{|c|}
\hline
\text{伯努利分布} \\
X \sim Bernoulli(\pi) \\
\mathbb{P}(X = x|\pi) = \pi^x(1-\pi)^{1-x}, \quad x = 0, 1 \\
E[X] = \pi \\
Var[X] = \pi(1-\pi) \\
M_x(t) = \pi e^t + \varrho \\
\hline
\end{array}
\tag{3.2}
$$

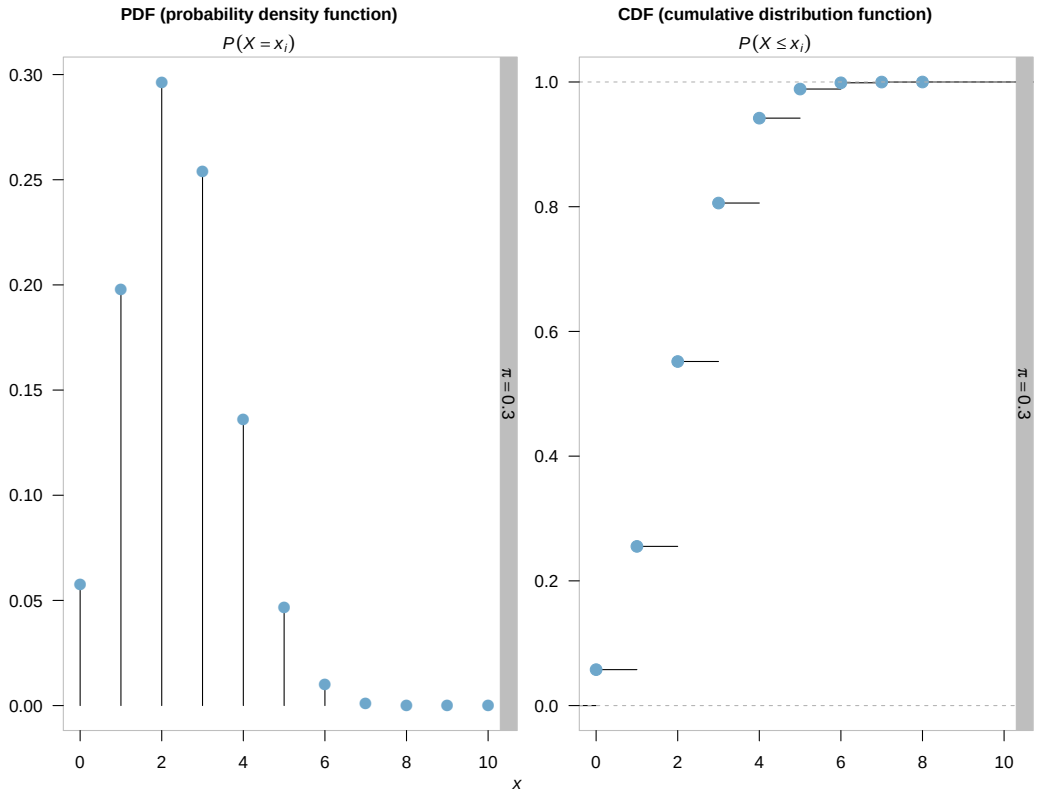
将伯努利试验 E 在相同条件下独立重复地进行 n 次，称这一串重复的独立试验为 n 重伯努利试验。在 n 重伯努利试验中，事件 A 恰好发生 k 次的概率为：

$$
\begin{array}{|c|}
\hline
\text{二项式分布} \\
X \sim Bin(n, \pi) \\
\mathbb{P}(X = x|n, \pi) = \left(\begin{array}{c} n \\ x \end{array} \right) \pi^x(1-\pi)^{n-x}, \quad x = 0, 1, 2, \cdots, n. \\
E[X] = n\pi \\
Var[X] = n\pi(1-\pi) \\
M_x(t) = (\pi e^t + \varrho)^n \\
\hline
\end{array}
\tag{3.3}
$$

因为该分布概率正好是二项式 $(\pi + \varrho)^n$ 展开式中出现 π^k 的那一项，所以称该随机变量服从参数为 (n, π) 的二项式分布（binominal Distribution）。

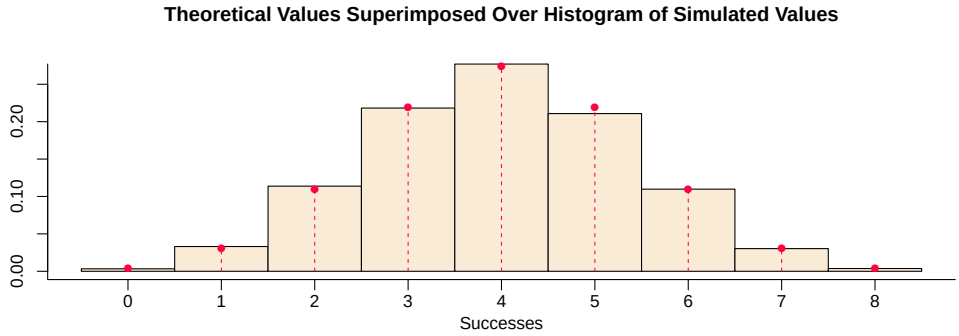
例如，下图展现的是 $n = 8$，$\pi = 0.3$ 时的二项分布的概率密度图和概率分布图：

```
acqr::Plot_Binomial_Density(size = 8, pi = 0.3)
```



下面一张图显示了二项分布的模拟数据和理论数据的对比情况:

```
acqr::Plot_Binomial_Simulate(size = 8, pi = 0.5)
```



3.2.3 泊松分布

在一定时间间隔之内质点出现的次数满足以下条件的计数过程称强度为 $\lambda, \lambda > 0$ 的泊松过程（Poisson proceses）：

1. 在不重叠的区间上出现的质点数具有独立性，即在区间 $(t, t+h], h > 0$ 上出现的质点数不受在区间 $(0, t]$ 出现的质点数的影响。

2. 对于充分小的 h，在区间 $(t, t+h], h > 0$ 出现 2 个或 2 个以上质点的概率与出现一个质点的概率相比可以忽略不计。

3. 在一个充分窄的区间内出现 1 个质点的概率与区间的宽度成正比，即在一个宽度为 h 的区间内出现一个质点的概率是 $\lambda h, \lambda > 0$。

泊松随机变量在时间段 t 内出现 k 个质点的概率服从参数为 λ 的泊松分布（Poisson Distribution），即：

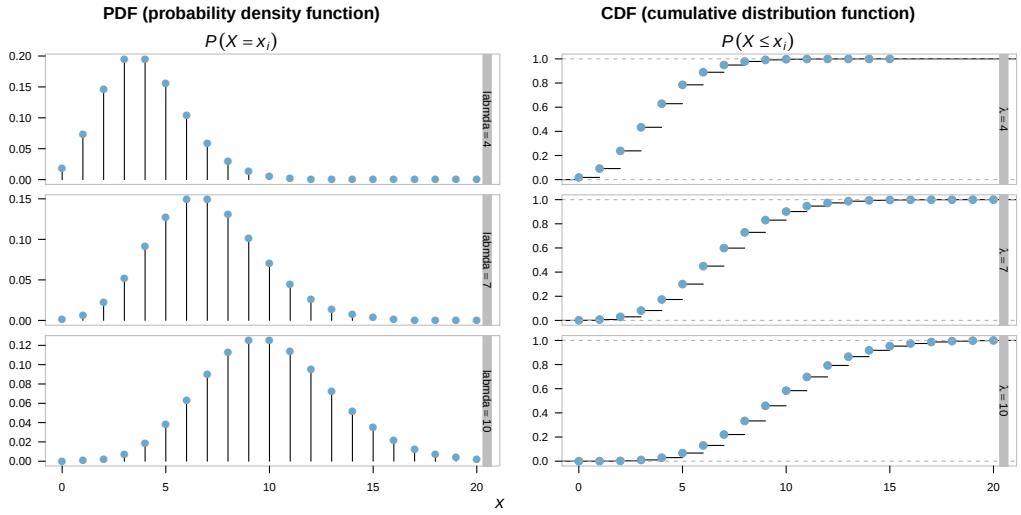
$$\mathbb{P}(X = x|\lambda t) = \frac{e^{-\lambda t}(\lambda t)^x}{x!} \quad x = 0, 1, \cdots, \quad \lambda > 0. \tag{3.4}$$

尤其，在一个单位时间段即 $t=1$ 内的泊松分布具有以下特征：

$$
\begin{aligned}
&\text{柏松分布}\\
&X \sim Pois(\lambda)\\
&\mathbb{P}(X=x|\lambda)=\frac{\lambda^x e^{-\lambda}}{x!} \quad x=0,1,\cdots, \quad \lambda>0.\\
&E[X]=\lambda\\
&Var[X]=\lambda\\
&M_x(t)=e^{\lambda(e^t-1)}
\end{aligned}
\tag{3.5}
$$

下面图形列出了几个泊松分布的概率密度图和概率分布图：

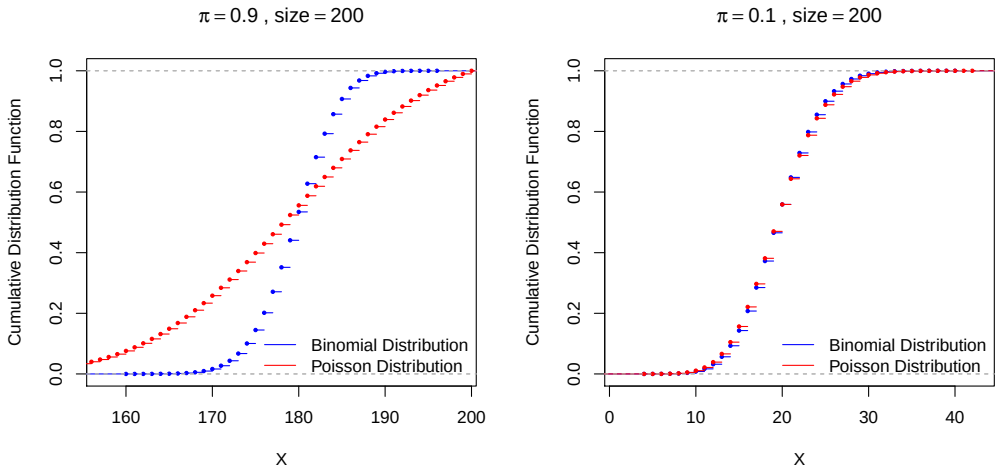
```
acqr::Plot_Poisson_Density()
```



柏松定理：设 $\lambda>0$ 是一个常数，$np_n=\lambda$，则对于任意一个固定的非负整数 k，有

$$
\lim_{n\to\infty}\left(\begin{array}{c} n \\ k \end{array}\right) p_n^k(1-p)^{n-k}=\frac{\lambda^k e^{-\lambda}}{k!} \tag{3.6}
$$

上述定理表明，当 n 很大，p 很小时，以 n, p 为参数的二项分布的概率值可以由参数为 $\lambda = np$ 的泊松分布的概率值近似。下面是两个泊松分布和二项式分布的累积分布概率的比较：

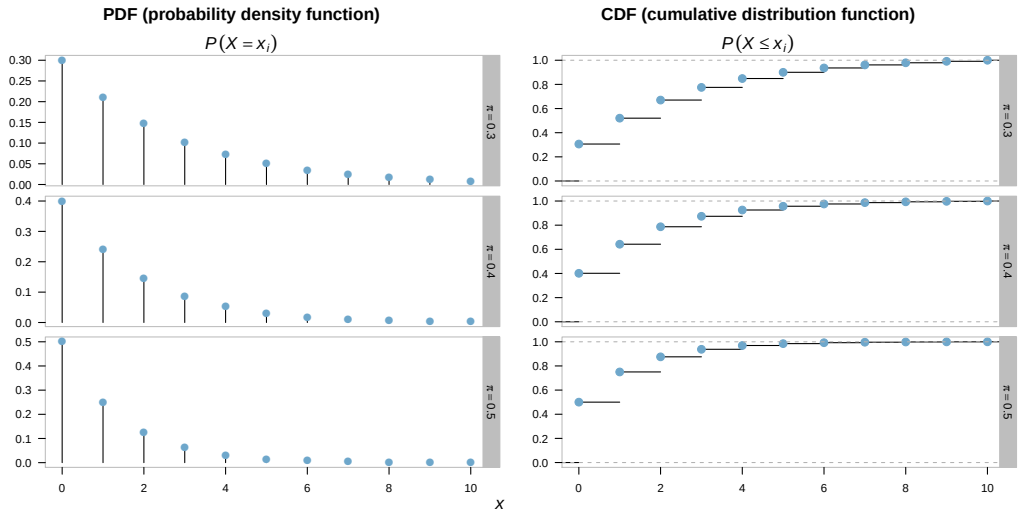


3.2.4 几何分布

在伯努利试验中，在得到第一次成功之前所经历的失败次数服从几何分布（Geometric Distribution）：

$$
\begin{aligned}
&\text{几何分布} \\
&X \sim Geo(\pi) \\
&\mathbb{P}(X = x|\pi) = \pi\varrho^x \quad x = 0, 1, \cdots, \quad \lambda > 0. \\
&E[X] = \frac{\varrho}{\pi} \\
&Var[X] = \frac{\varrho}{\pi^2} \\
&M_x(t) = \frac{\pi}{1 - \varrho e^t}
\end{aligned}
\tag{3.7}
$$

```
acqr::Plot_Geometric_Density()
```

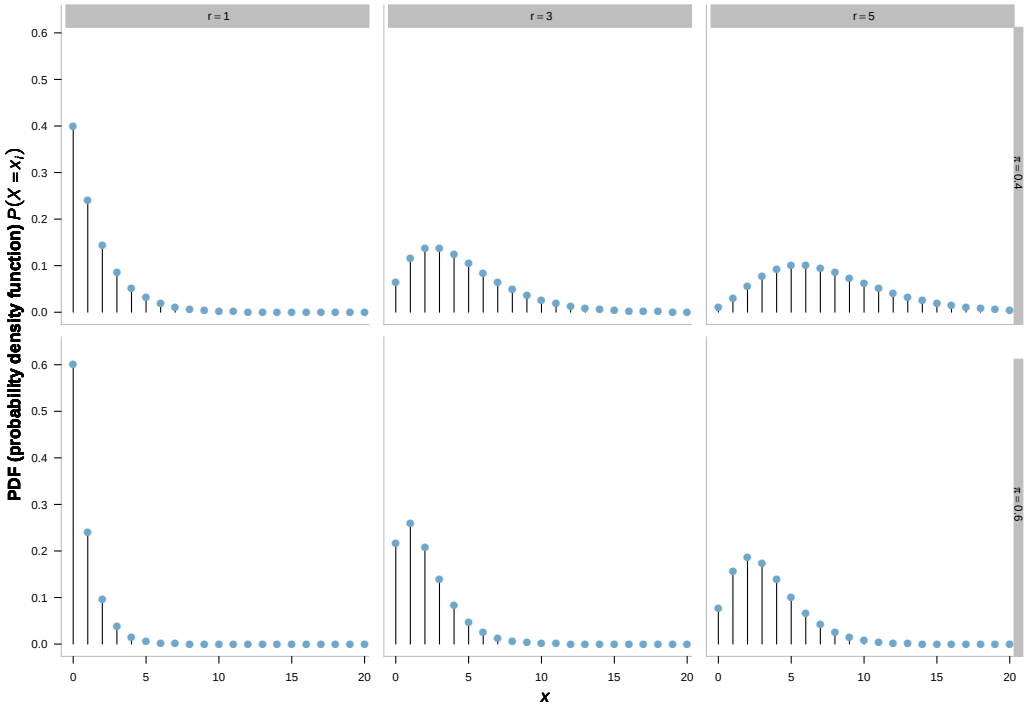


3.2.5 负二项式分布

负二项式分布（Negative Binomial Distribution）指所有到成功 r 次时即终止的独立试验中，失败次数 X 的分布：

$$
\begin{aligned}
&\text{负二项式分布} \\
&X \sim NB(r, \pi) \\
\mathbb{P}(X = x|r, \pi) &= \left(\begin{array}{c} x+r-1 \\ r-1 \end{array} \right) \pi^r \varrho^x, \quad x = 0, 1, 2, \cdots, n. \\
E[X] &= r\frac{\varrho}{\pi} \\
Var[X] &= r\frac{\varrho}{\pi^2} \\
M_x(t) &= \pi^r (1 - \varrho e^t)^{-r}
\end{aligned}
\tag{3.8}
$$

例如，下面是取不同 π 值和不同 r 值时负二项式分布的概率密度图：



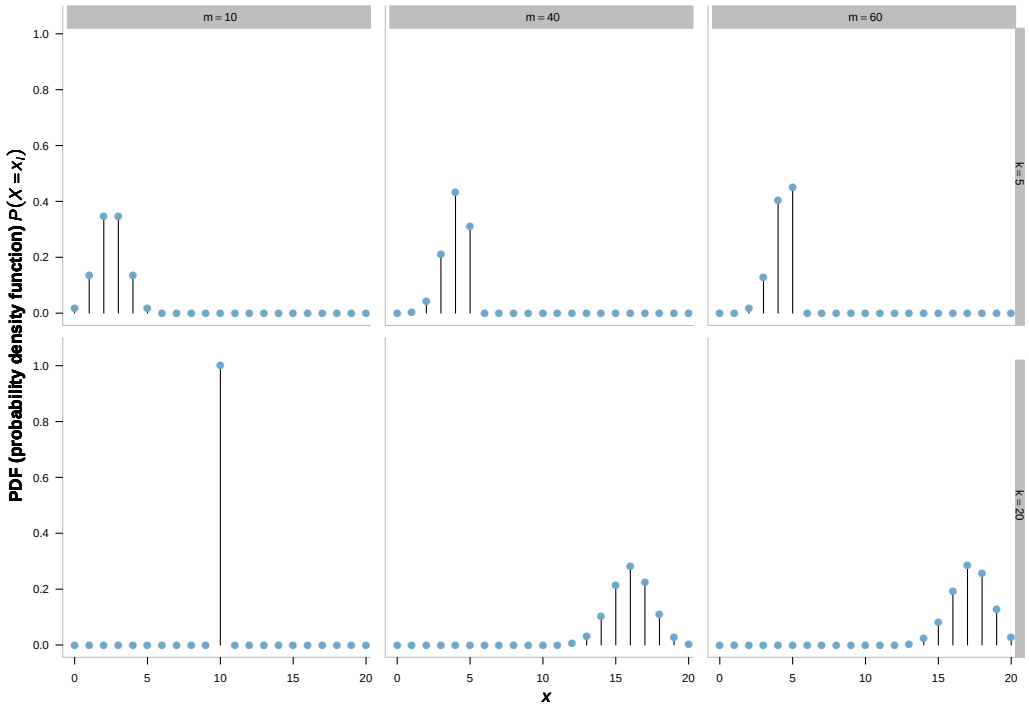
3.2.6 超几何分布

一个二分分布总体中共有 N 个商品，其中 m 个商品合格，n 个商品不合格，即 $N = m+n$，那么从这个整体中抽取 k 个商品，其中 x 个合格商品，

$k-x$ 个不合格商品的概率服从超几何分布（hypergeometric distribution）：

$$\begin{array}{c} \text{超几何分布} \\ X \sim Hyper(m,n,k) \\ \mathbb{P}(X=x|m,n,k) = \dfrac{\begin{pmatrix} m \\ x \end{pmatrix}\begin{pmatrix} n \\ k-x \end{pmatrix}}{\begin{pmatrix} N \\ k \end{pmatrix}}. \\ E[X] = \dfrac{m \times k}{N} \\ Var[X] = \dfrac{m \times n \times k \times (N-k)}{N^2 \times (N-1)} \end{array} \tag{3.9}$$

例如，下面是当 $n=10$ 时，取不同 m 值和不同 k 值时超几何分布的概率密度图：



注意，当 $\frac{k}{N}$ 很小时（≤ 0.10）时，一个超几何分布与一个 $n = k$ 和 $\pi = \frac{m}{N}$ 的分布很相似。

3.3 连续型单变量

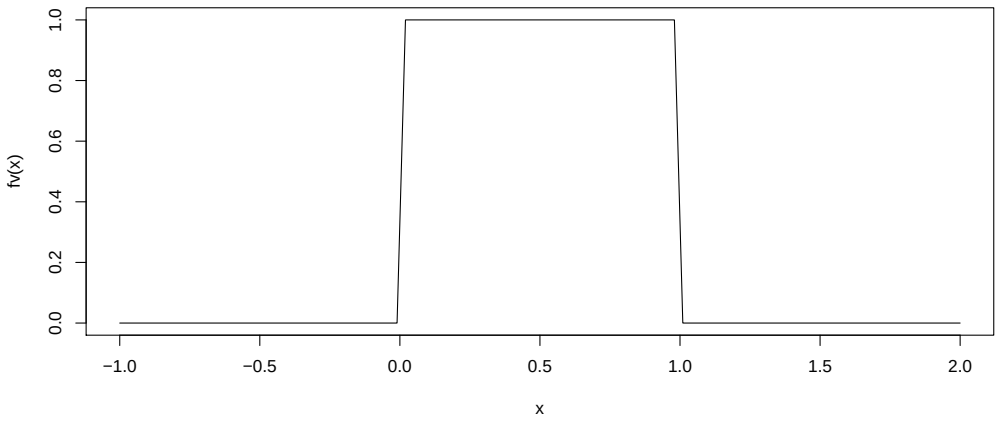
3.3.1 连续型均匀分布

若连续型随机变量 X 具有如下概率密度，则称 X 在区间 (a, b) 上服从均匀分布（Continuous Uniform Distribution）：

连续型均匀分布

$$
\begin{aligned}
X &\sim Unif(a, b) \\
f(x|a, b) &= \frac{1}{b-a}, \quad a \leq x \leq b \\
E[X] &= \frac{b+a}{2} \\
Var[X] &= \frac{(b-a)^2}{12} \\
M_X(t) &= \begin{cases} \frac{e^{tb}-e^{ta}}{t(b-a)} & \text{if } t \neq 0 \\ 1 & \text{if } t = 0 \end{cases}
\end{aligned}
\tag{3.10}
$$

```
ff <- function(x, a = 0, b = 1) {
    if (a > b) {
            stop("`a' should not be bigger than `b'")
        } else if (x >= a && x <= b) {
                1 / (b - a)
                } else 0
}
fv <- Vectorize(ff)
curve(fv, -1, 2)
```




```
polygon()
```

```
Error in xy.coords(x, y, setLab = FALSE): argument "x" is missing,
with no default
```

3.3.2 指数分布

3.3.3 gamma 分布

3.3.4 生存分布

3.3.5 韦伯分布

3.3.6 Beta 分布

3.3.7 正态分布

參考文獻

Sarkar, D. (2008). *Lattice: Multivariate data visualization with r*. Springer.

Wickham, H. (2009). *Ggplot2: Elegant graphics for data analysis*. New York, NY: Springer.