R-7

郑泽靖　zzjstat2023@163.com

北京师范大学统计学院

集中趋势的分析

```
1、样本均值  mean()
2、样本中位数 median()
3、样本分位数 quantile(x,a,type=5)
type在1:9取值，代表9种计算分位数的方法
4、众数
#连续型变量样本数据的众数
x=runif(100,5,10)
hist(x)
y <- hist(x,plot=F) #计算x的分组数据统计结果
maxID <- which.max(y$counts) #计算最大频数所在区间序号
mean(y$breaks[maxID:(maxID+1)])#计算第maxID区间的中心
#离散型变量样本数据的众数
x=sample(1:10,100,T)
y <- table(x) #计算x的分组数据统计结果
names(which.max(y)) #计算最大频数
```

集中趋势的分析

```
5.方差： var(x)
x<-c (42,55,64,70,75,78,80,82,82,82,
 85,85,85,85,88,90,90,92,95,99)
y<-c(39,52,61,68,72,76,77,78,79,78,
 83,83,81,81,85,87,86,91,91,98)
var(x)
var(y)
6.标准差： sd (x)
sd(x)
7.标准化 scale(x)
```

QQ 图

使用 `qqnorm` 和 `qqline` 绘制正态 QQ 图。当样本来自正态分布时，散点应近似分布在一条直线附近。

QQ 图的做法如下：设有 n 个观测值 $y_1, y_2, \ldots, y_n$，从小到大排列。对于样本来自 $N(\mu, \sigma^2)$，有：

$$y_i \approx \mu + \sigma x_i$$

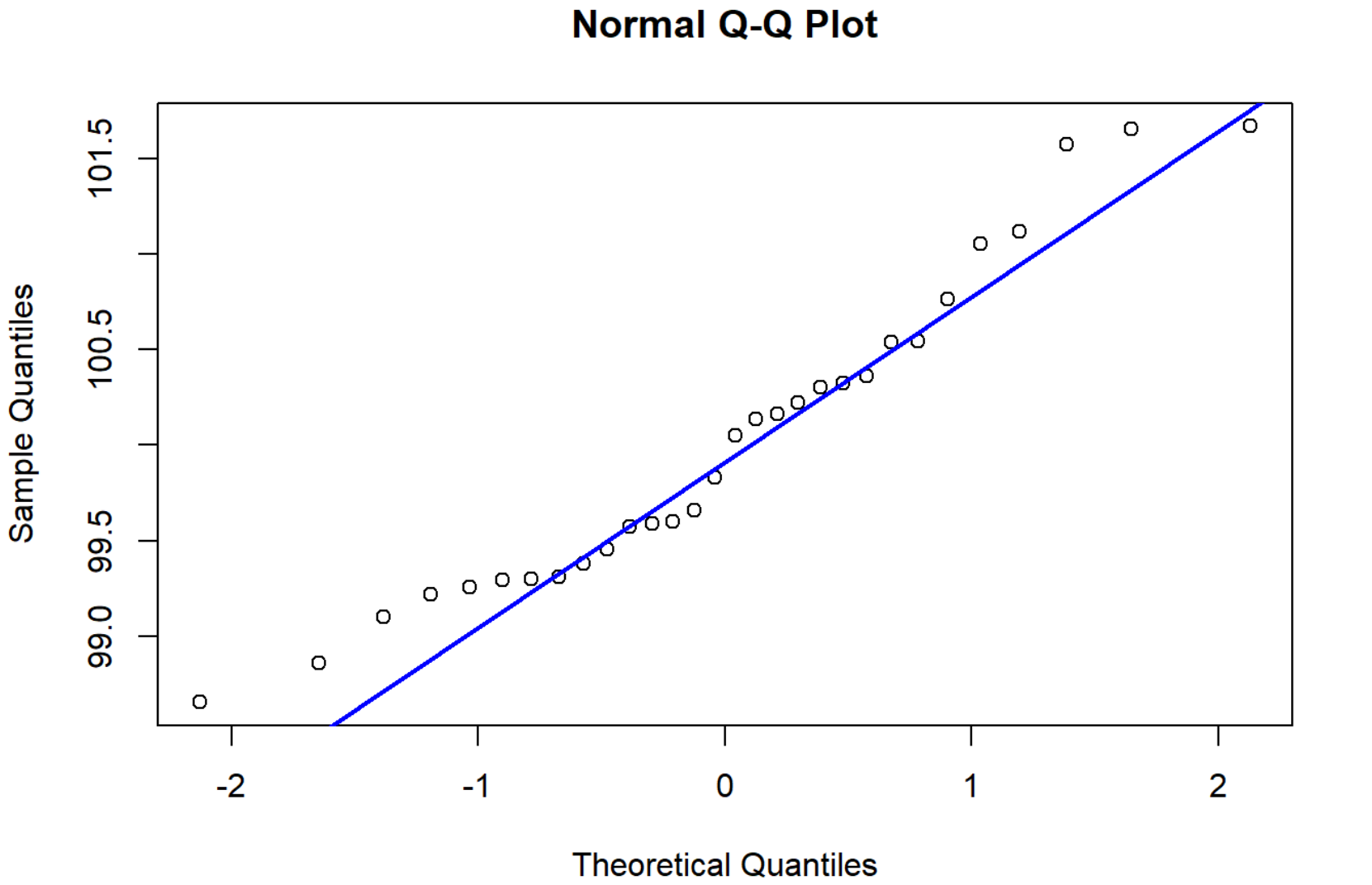
其中 x_i 是标准正态分布的分位数。

R 代码示例：

```r
qqnorm(x)
qqline(x, lwd=2, col='blue')
```

散点越集中在对角线附近，数据越可能服从正态分布。

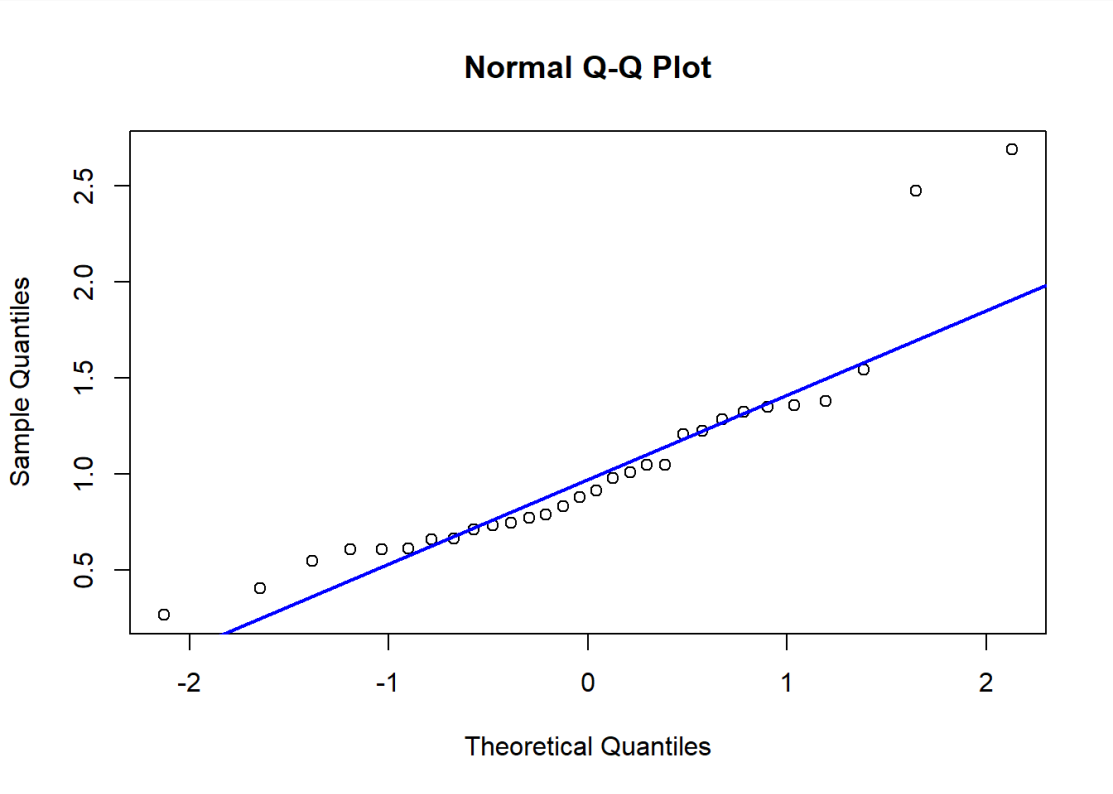
QQ 图


Normal Q-Q Plot
Sample Quantiles
99.0
99.5
100.5
101.5
-2
-1
0
1
2
Theoretical Quantiles

QQ 图

下面的程序模拟对数正态数据，并作正态 QQ 图:

```
z <- 10^rnorm(30, mean=0, sd=0.2)
qqnorm(z)
qqline(z, lwd=2, col='blue')
```



五数概括

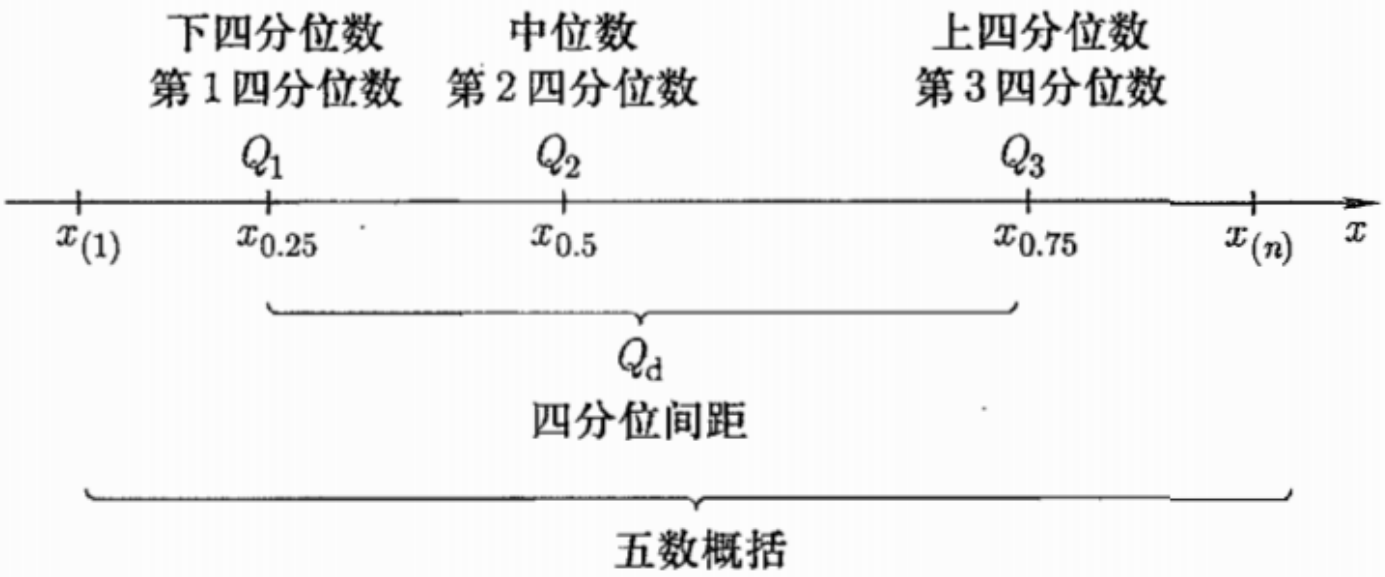


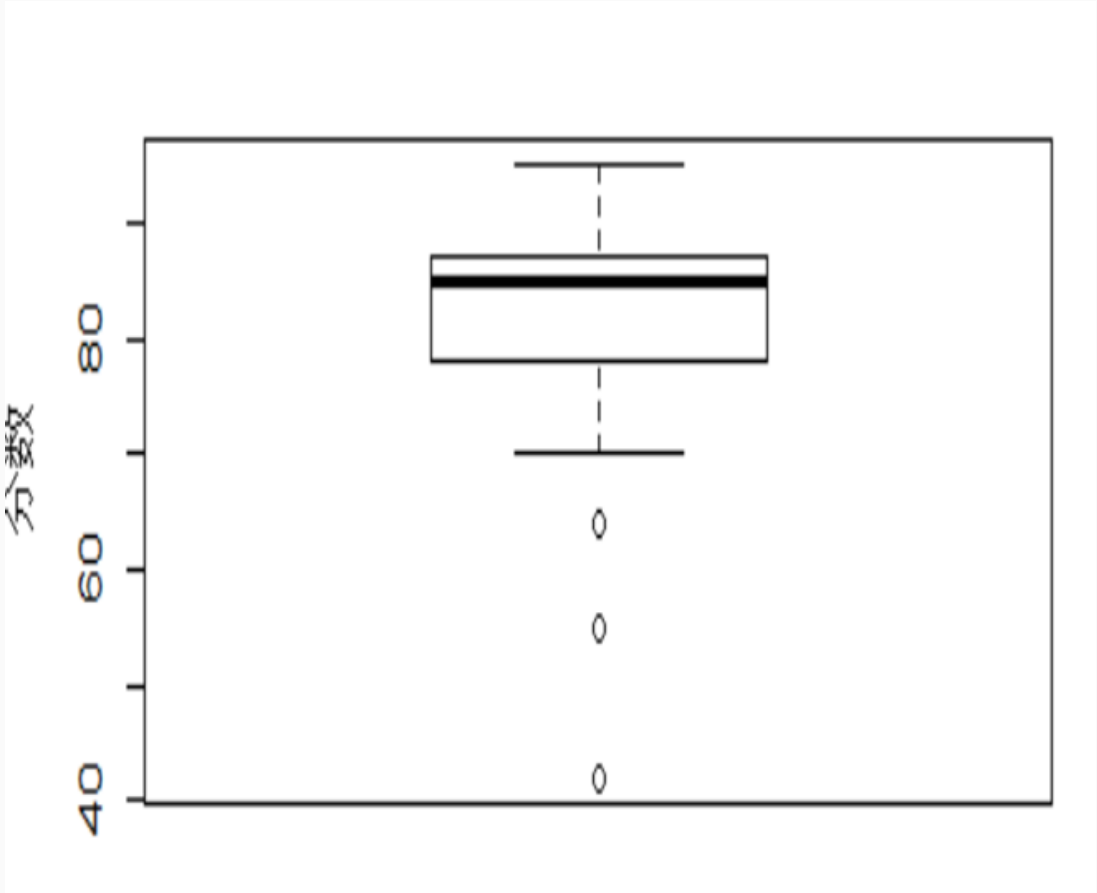
图 4.25　五数概括

盒形图

```
x <-c(42,55,64,70,75,78,80,82,82,82,85,85,85,85,85,88
,90,90,92,95,87)
fivenum(x) #计算五数概括：四分位数，最小值，最大值
boxplot(x,range=1.5,horizontal=F,ylab="分数")
```

其中：range: 默认 1.5，表示上、下两条须线的长度不超过四分位间距（上四分位数-下四分位数）的 1.5 倍 horizontal=T，则为水平的盒形图 x: 也可以是数据框，对其中的每一个变量画盒形图

盒形图



盒形图

使用 boxplot 函数可以在同一坐标系下绘制来自不同总体的样本数据的盒形图，方便比较它们的分布特征和离群数据情况。只需将不同的样本数据放入一个数据框，然后将该数据框作为 boxplot 的输入变量。

例如，当数据框 xy 的各列为不同总体的样本数据时，使用以下代码绘制盒形图：

```
boxplot(xy)
例：
xy <- data.frame("甲班"=x, "乙班"=y)
boxplot(xy, range=1.5, ylab="分数")
```

盒形图

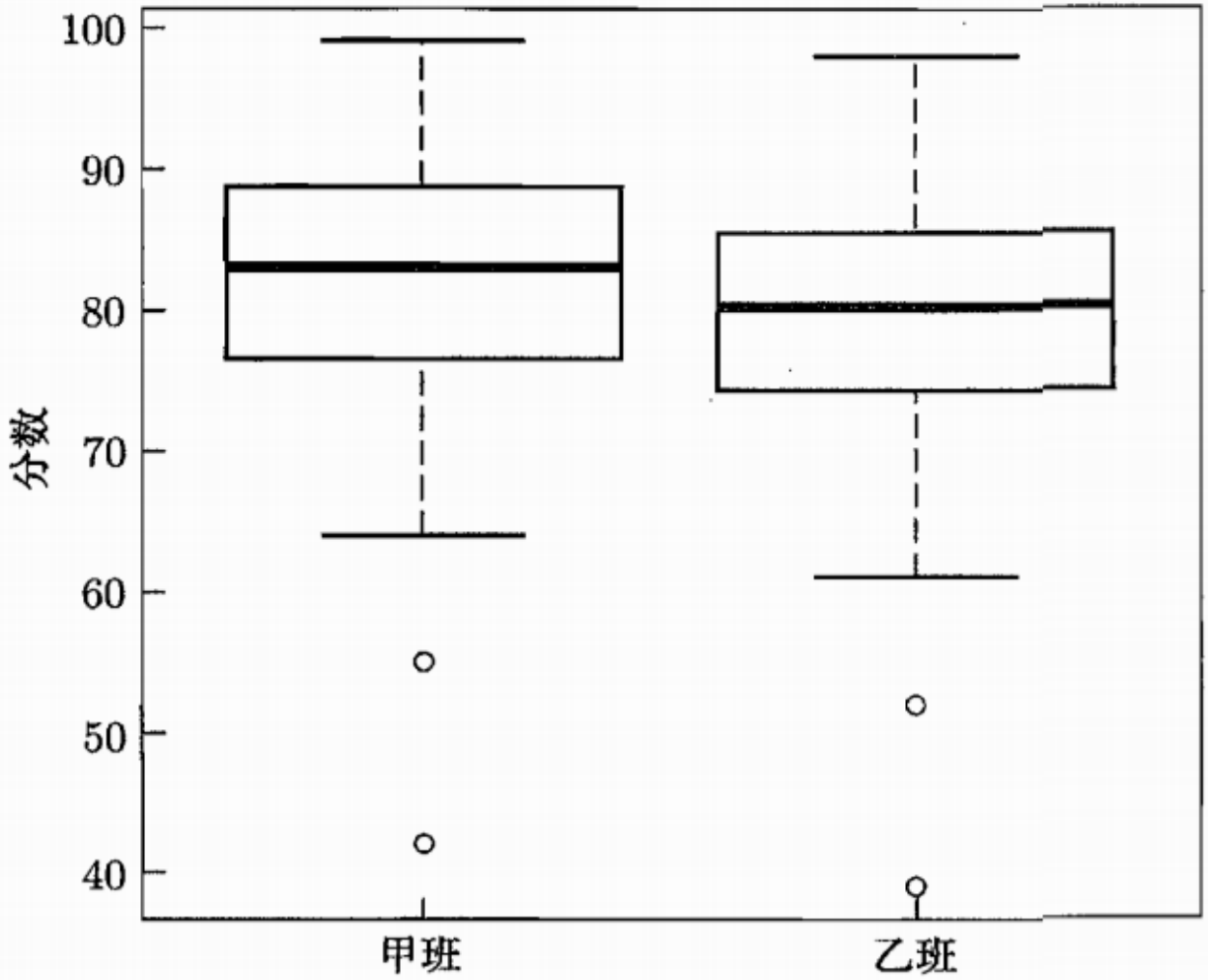


图 4.28　两个班级学生数学成绩的盒形图

apply 函数

```r
x <- matrix(rnorm(1000*5),1000,5)
#生成1000组容量为5的样本，
每个样本都服从标准正态分布，排列成1000行5列的矩阵。
apply(x,MARGIN=1,FUN=mean)
#对x的每一行求均值
```

apply 函数能够极大减少 for 循环的使用:

```r
apply(X, MARGIN, FUN, ...)
```

- x: 运算对象：数组、矩阵、数据框，数据至少是二维的
- MARGIN: 取 1，2，c(1,2) 分别表示对运算对象的行、列、同时对二者运算
- FUN：指定运算函数，如 mean 表示求均值

apply 函数

```
> (my.matrx <- matrix(c(1:10, 11:20, 21:30),
nrow = 10, ncol = 3))
      [,1] [,2] [,3]
 [1,]    1   11   21
 [2,]    2   12   22
 [3,]    3   13   23
 [4,]    4   14   24
 [5,]    5   15   25
 [6,]    6   16   26
 [7,]    7   17   27
 [8,]    8   18   28
 [9,]    9   19   29
[10,]   10   20   30
```

apply 函数

```
#对每一行求和
> apply(my.matrx, 1, sum)
 [1] 33 36 39 42 45 48 51 54 57 60
#计算每一列的长度
> apply(my.matrx, 2, length)
[1] 10 10 10
#传递自定义匿名函数
> apply(my.matrx, 2, function (x) length(x)-1)
[1] 9 9 9
#使用定义在外部的函数
> st.err <- function(x){
+     sd(x)/sqrt(length(x))
+ }
> apply(my.matrx,2, st.err)
[1] 0.9574271 0.9574271 0.9574271
```

apply 函数

```
> (my.matrx2 <- apply(my.matrx, 1:2, function(x) x+3))
      [,1] [,2] [,3]
 [1,]    4   14   24
 [2,]    5   15   25
 [3,]    6   16   26
 [4,]    7   17   27
 [5,]    8   18   28
 [6,]    9   19   29
 [7,]   10   20   30
 [8,]   11   21   31
 [9,]   12   22   32
[10,]   13   23   33
```

最大似然估计

R 函数 optim()、nlm()、optimize() 可以用来求函数极值，因此可以用来计算最大似然估计。optimize() 只能求一元函数极值。
正态分布最大似然估计有解析表达式。作为示例，用 R 函数进行数值优化求解。

对数似然函数为：

$$\ln L\left(\mu, \sigma^2\right) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum\left(X_i - \mu\right)^2$$

定义 R 的优化目标函数为上述对数似然函数去掉常数项以后乘以，求其最小值点。

最大似然估计

目标函数为：

```
objf.norm1 <- function(theta, x){
  mu <- theta[1]
  s2 <- exp(theta[2])
  n <- length(x)
  res <- n*log(s2) + 1/s2*sum((x - mu)^2)
  res
}
```

其中 θ_1 为均值参数 μ θ_2 为方差参数 σ^2 的对数值。x 是样本数值组成的 *R* 向量。

optim 函数

可以用 optim 函数来求极小值点。下面是一个模拟演示:

```
n<-30
mu0 <- 20
sigma0 <- 2
set.seed(1)
x <- rnorm(n, mu0, sigma0)
theta0 <- c(0,0)
ores <- optim(theta0, objf.norm1, x=x)
print(ores)
theta <- ores$par
mu <- theta[1]
sigma <- exp(0.5*theta[2])
```

optim 函数

```
## $par
## [1] 20.166892  1.193593
##
## $value
## [1] 65.83709
##
## $counts
## function gradient
##      115       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## 真实mu= 20  公式估计mu= 20.16492
## 数值优化估计mu= 20.16689
## 真实sigma= 2 公式估计sigma= 1.817177
## 数值优化估计sigma= 1.816291
```

optimize()

函数 optimize() 进行一元函数数值优化计算。例如，在一元正态分布最大似然估计中，在对数似然函数中代入 $\mu = \bar{x}$，目标函数：

$$h\left(\sigma^2\right) = \ln\left(\sigma^2\right) + \frac{1}{\sigma^2}\sum\left(X_i - \bar{X}\right)^2$$

下面的例子模拟计算 σ^2 的最大似然估计：

```
n<-30
mu0 <- 20;  sigma0 <- 2;  set.seed(1)
x <- rnorm(n, mu0, sigma0); mu <- mean(x)
ss <- sum((x - mu)^2)/length(x)
objf <- function(delta, ss) log(delta) + 1/delta*ss
ores <- optimize(objf, lower=0.0001,
                 upper=1000, ss=ss)
delta <- ores$minimum;sigma <- sqrt(delta)
print(ores)
```

```
## $minimum
## [1] 3.30214
##
## $objective
## [1] 2.194568
##
## 真实sigma= 2 公式估计sigma= 1.817177
## 数值优化估计sigma= 1.817179
```

该函数的计算结果会存在列表中，是有两个分量的列表变量:

```
#第一个分量$maximum: 最优的参数取值
#第二个分量$objective: 最优（最大或最小）的函数值
```

矩估计

1、矩估计矩分为
(1) 原点矩 $\mu_k = E\left(X^k\right)$, 均值: $\mu = E(X), k = 1$
(2) 中心矩 $v_k = E(X - E(X))^k$, 方差: $v_2 = E(X - E(X))^2$
2. 矩估计即
(1) 用样本矩估计总体矩
(2) 用样本矩的函数估计总体矩的函数
$\rightarrow$ 把所有的 E 都变成求均值
例子: 用样本均值 $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ 估计总体均值 u

练习

1. 生成 100 个来自 $x \sim N(2, 4)$ 的随机变量值，计算其均值和标准差。重复试验 100 次，绘制均值和标准差的 QQ 图观察其结果.

示例

```r
# 进行100次实验
n_trials <- 100
means <- numeric(n_trials)  # 用于存储均值
sds <- numeric(n_trials)    # 用于存储标准差

# 重复试验100次
for (i in 1:n_trials) {
  sample_data <- rnorm(100, mean = 2, sd = 2)  # 生成100个来
  means[i] <- mean(sample_data)  # 计算均值
  sds[i] <- sd(sample_data)      # 计算标准差
}

# 绘制均值的QQ图
par(mfrow=c(1, 2))  # 设置图形窗口为1行2列
qqnorm(means, main = "均值的QQ图")
qqline(means, col='blue', lwd=2)

# 绘制标准差的QQ图
qqnorm(sds, main = "标准差的QQ图")
qqline(sds, col='red', lwd=2)
```

Thanks!