

基于 R 软件实现随机抽样及其应用

张效嘉¹ 胡良平^{1 2*}

(1. 军事医学科学院生物医学统计学咨询中心, 北京 100850;

2. 世界中医药学会联合会临床科研统计学专业委员会, 北京 100029

* 通信作者: 胡良平, E-mail: lphu812@sina.com)

【摘要】 本文的目的是使读者快速掌握用 R 软件产生服从多种分布规律的随机数和多种随机抽样的方法。通过介绍随机数生成器的概念、对应的函数、实际生成服从正态分布和均匀分布的随机数, 读者很容易获得基于 R 软件产生所需要的随机数; 通过介绍有放回与无放回随机抽样、系统与分层随机抽样、整群与 bootstrap 随机抽样的概念和具体实现, 使读者轻松地基于 R 软件实现多种随机抽样。事实表明: R 软件易于获取、易学易用; R 软件功能强大、适用面宽, 可方便快捷地解决试验设计中的随机抽样问题。

【关键词】 R 软件; 随机数生成器; 正态分布; 均匀分布; 随机抽样; bootstrap 随机抽样

中图分类号: R195.1

文献标识码: A

doi: 10.11886/j.issn.1007-3256.2016.06.002

The realization of the random sampling and its application based on R software

Zhang Xiaojia¹, Hu Liangping^{1 2*}

(1. Consulting Center of Biomedical Statistics, Academy of Military Medical Sciences, Beijing 100850, China;

2. Specialty Committee of Clinical Scientific Research Statistics of World Federation of Chinese Medicine Societies, Beijing 100029, China

* Corresponding author: Hu Liangping, E-mail: lphu812@sina.com)

【Abstract】 The paper aims to make it convenient for readers to utilize R software to generate random number from various distributions and implement varieties of random sampling methods. The paper introduced the conception and corresponding functions of random number generators. We presented cases of generating random number from both normal distribution and uniform distribution. Consequently, readers may easily generate random number by R software. The paper also introduced the conception and realization of random sampling both with and without replacement, systematic random sampling, stratified random sampling, cluster random sampling and bootstrap random sampling. Therefore, readers may use R software to realize varieties of random sampling methods. The fact indicated that R software may be simply obtained and utilized. R software may be used to solve random sampling problem in experimental designs conveniently, due to its powerful function and wide application.

【Key words】 R software; Random number generator; Normal distribution; Uniform distribution; Random sampling; Bootstrap random sampling

1 产生随机数^[1-3]

1.1 随机数及其种类

1.1.1 概述

所谓随机数, 就是它们在数量大小和先后顺序等方面, 出现的规律是不确定的。然而, 人们还可以在一定的前提条件限制下, 去生成一系列的随机数, 它们出现的先后顺序是随机的, 但它们在整体上却是服从某种特定规律的。

由此可知, 当人们将“前提条件”设定为“均匀分布”, 就可生成均匀分布随机数; 当人们将“前提条件”设定为“正态分布”, 就可生成正态分布随机数; ……; 以此类推, 还可生成很多服从其他特定分

布规律的随机数。

1.1.2 随机数生成器种类

在计算机上, 生成随机数需要基于一定的算法。不同的算法会生成不同的随机数。R 提供了多种随机数生成器 (random number generators, RNG), 默认的 RNG 是由 Makoto Matsumoto 与 Takuji Nishimura 于 1997 年 - 1998 年提出的 RNG, 即 Mersenne twister 随机数生成器。该 RNG 的循环周期为 $2^{19937} - 1$ 。R 还提供了其他几种 RNG, 通过在 R 的控制台发送命令 “help(RNGkind)”, 可以获得有关的详细信息。

若用户想把默认的 RNG 更改为自己指定的某种其他的 RNG, 可通过在 R 的控制台发送命令 “RNGkind(kind = “new_rng”)”。注意: 这里的 “new_rng” 只能是下列几种备选内容: Mersenne(默认项)、Wich、

项目基金: 国家高技术研究发展计划课题资助(2015AA020102)

Mars, Super, Knuth - TAOCP - 2002、Knuth - TAOCP、L'Ecuyer - CMRG。

1.1.3 确保每批生成的随机数相同

一般来说,生成随机数的函数每批生成的随机数是不同的。但有时,人们需要生成相同批次的随机数(如在临床试验设计和模拟试验时),此时,可以在运行某种随机数函数时先运行一下 `set.seed(n)` 函数(这里 `n` 为一个确定的正整数,如 1 或 2 等)。

现以生成均匀分布随机数的 `runif()` 函数为例,在运行此随机数函数之前先运行 `set.seed()` 函数,可使 `runif()` 函数每批生成的随机数相同(生成的随机数的长度应相同)。

```
> set.seed(1)
```

```
> runif(10)
```

以上为第一次运行,要求生成在 $[0, 1]$ 区间内且服从均匀分布的 10 个随机数。

```
[1] 0.26550866 0.37212390 0.57285336
0.90820779 0.20168193 0.89838968
```

```
[7] 0.94467527 0.66079779 0.62911404
0.06178627
```

以上是第一次运行输出的 10 个随机数。

```
> set.seed(1)
```

```
> runif(10)
```

以上为第二次运行,仍要求生成在 $[0, 1]$ 区间内且服从均匀分布的 10 个随机数(与前面的 10 个随机数完全相同,此处从略)。

显然,以上两次运行,生成的在 $[0, 1]$ 区间内且服从均匀分布的 10 个随机数完全相同。

1.2 用 R 生成正态分布随机数

【例 1】试生成 100 个服从均值为 0、标准差为 1 的标准正态分布的随机数。

解答:在 R 中,使用 `rnorm()` 函数可以生成服从正态分布的随机数。其语法规则是: `rnorm(n, mean = k1, sd = k2)`,其中,参数 `n` 代表需要生成的随机数的个数、`k1` 为用户指定的均值、`k2` 为用户指定的标准差。当 `k1 = 0`、`k2 = 1` 时,就是希望生成服从标准正态分布的 `n` (`n` 必须为一个具体的正整数) 个随机数。 `rnorm(n, mean = 0, sd = 1) = rnorm(n, 0, 1) = rnorm(n)`。

```
> r_number1 <- rnorm(100, 0, 1); r_number1
```

以上语句的目的是生成 100 个服从均值 = 0、标准差 = 1 的标准正态分布的随机数并将其赋值给变

量(本质上是一个向量) `r_number`,然后,将其输出(输出结果从略)。

1.3 用 R 生成均匀分布随机数

在 R 中,使用 `runif()` 函数可以生成服从均匀分布的随机数。其语法规则是: `runif(n, min = k1, max = k2)`,其中,参数 `n` 代表需要生成的随机数的个数、`k1` 为用户指定的服从均匀分布的随机数的下限、`k2` 为用户指定的服从均匀分布的随机数的上限。若省略参数 `min`、`max`,则默认生成 $[0, 1]$ 上的均匀分布随机数。

【例 2】试生成 1000 个服从下限为 2、上限为 12 的均匀分布的随机数,并用直方图展示它们。

解答:为实现题中的目标,可使用下面的语句。

```
> x <- runif(1000, min = 2, max = 12)
```

```
> hist(x, prob = T, main = "uniform distribution
(min = 2, max = 12)")
```

第一句的目的是生成 1000 个服从下限为 2、上限为 12 的均匀分布的随机数。

第二句的目的是绘制这 1000 个随机数的直方图(无法绘制非标准均匀分布密度函数曲线)。产生的直方图从略。

1.4 用 R 生成其他分布随机数

1.4.1 生成指数分布随机数

生成指数分布随机数的函数为 `rexp()`,其语法规则是: `rexp(n, lamda = 1/mean)`,这里 `n` 代表生成的随机数的个数,而 `lamda` 代表随机数的平均值的倒数,`mean` 代表随机数的平均数。

```
> x <- rexp(1000, 1/110); x
```

以上语句的目的是生成 1000 个服从均值为 110 的指数分布的随机数并将其输出(输出结果此处从略)。

1.4.2 生成贝塔分布随机数

生成贝塔分布随机数的函数为 `rbeta()`,其语法规则是: `rbeta(n, a, b)`,这里 `n` 代表生成的随机数的个数,而 `a` 和 `b` 代表该分布的两个非负的实数($a > 0$, $b > 0$)。

```
> x <- rbeta(100, 2, 8); x
```

以上语句的目的是生成 100 个服从参数 `a = 2`、参数 `b = 8` 的贝塔分布的随机数并将其输出(输出结果此处从略)。

1.4.3 生成伽玛分布随机数

生成伽玛分布随机数的函数为 `rgamma()` ,其语法规则是: `rgamma(n, shape, rate = 1, scale = 1/rate)` ,这里 `n` 代表生成的随机数的个数,而 `shape` 代表该分布的形状参数、`scale` 代表尺度参数,这两个参数都应该为非负实数(`shape > 0`、`scale > 0`)。注意: `rate` 与 `scale` 两个中只能给定一个的具体值。

```
> x <- rgamma(100, 1, 4); x
```

以上语句的目的是生成 100 个服从形状参数 `shape = 1`、尺度参数 `scale = 1/4` 的伽玛分布的随机数并将其输出(输出结果此处从略)。

说明:在 R 中,还可生成许多服从其他分布的随机数,因篇幅所限,此处从略。

2 随机抽样^[1-4]

2.1 概述

从给定的一个总体中抽取一定数目的样品,称为抽样。若总体中的所有样品被放在一个黑色的袋子中,样品的形状、大小、颜色和重量是相同的,只是上面的编号不同。当人们从中抽取若干个样品时,就称为随机抽样。这里的“随机”是指被抽取的样品是哪些,事先是不知道的。若从 `n` 个样品(假定它们构成一个总体)中一次性抽取 `m` (`m < n`) 个或一个一个地抽取(每次只抽取,不放回)连续共抽取 `m` 次,都称为无放回随机抽样;若从 `n` 个样品(假定它们构成一个总体)中一个一个地抽取(每次抽取后记录其编号,并将其放回)连续共抽取 `m` 次,就称为有放回随机抽样。

在 R 中,可以使用 `sample()` 函数来实现无放回与有放回随机抽样。其语法规则如下:

```
Sample(x, n, replace = F 或 T, prob = NULL)
```

`x` 代表总体向量,可以是数据、字符、逻辑向量;`n` 代表样本含量;`replace = F`,代表无放回随机抽样(这是默认的),`replace = T`,代表有放回随机抽样;`prob` 可以设置各个抽样单元不同的入样概率,进行不等概率抽样。

2.2 有放回随机抽样举例

【例 3】假定掷一枚质地均匀的硬币,出现正面用“H”表示、出现反面用“B”表示,现重复抛硬币 20 次,显示有放回随机抽样的结果。

解答:在 R 中使用下面的语句就可实现前述的

目的。

```
> sample(c("B", "H"), 20, rep = T)
```

```
[1] "B" "B" "H" "B" "H" "B" "H" "B" "H" "H" "B" "H" "H" "H" "H" "H" "H" "H" "H" "H"
```

```
[19] "B" "B"
```

以上是 20 次试验的结果,每次试验的结果要么是“B”、要么是“H”。其中,“B”出现了 9 次、“H”出现了 11 次。

【例 4】假定掷一枚质地均匀的骰子(6 个面上分别有 1、2、3、4、5、6 个点),现重复抛骰子 20 次,显示有放回随机抽样的结果。

解答:在 R 中使用下面的语句就可实现前述的目的。

```
> sample(c(1:6), 20, rep = T)
```

```
[1] 3 2 4 1 5 1 6 1 6 4 3 6 3 6 5 6 4 4 2 2
```

以上是 20 次试验的结果,每次试验的结果是 1 到 6 六个数字中的一个出现。这批试验的结果表明,1 出现了 3 次、2 出现了 3 次、3 出现了 3 次、4 出现了 4 次、5 出现了 2 次、6 出现了 4 次。

2.3 无放回随机抽样举例

【例 5】现有编号为 1~200 的 200 位受试对象,希望从中随机抽取 20 位。试显示被随机抽取的 20 位受试者的编号。

解答:在 R 中使用下面的语句就可实现前述的目的。

```
> sample(200, 20, rep = F)
```

```
[1] 180 47 91 124 22 141 36 186 193 8 53 67 113 68 154 143 108 59
```

```
[19] 123 71
```

以上是从 1~200 个编号中无放回地随机抽取的 20 个编号。

2.4 系统随机抽样举例

在 R 中,要想实现分层随机抽样或整群随机抽样,需从 R 软件中下载 `sampling` 包,分别调用其中的 `strata()` 函数与 `cluster()` 函数;与此同时,这个包中还有 `getdata()` 函数,可以读取随机抽样的全部样品对应的全部信息(否则,只有与抽样有关的部分信息)。

但是在 R 中,笔者目前尚未找到专门用于系统随机抽样的函数。虽然在下面的分层随机抽样的 `strata()` 函数与整群随机抽样的 `cluster()` 函数中,都涉及一个参数,即“`method =`”,在等号后面可以输

入四种抽样方法之一,即 srswor(无放回随机抽样)、srswr(有放回随机抽样)、poisson(泊松随机抽样)和 systematic(系统随机抽样),但这四种具体的抽样方法都是在前面的分层随机抽样或整群随机抽样基础之上的,而不是直接实施这四种具体随机抽样的。故此处暂不能进行系统随机抽样,若用户今后从 R 软件中发现了某个包中的某些函数可直接进行系统随机抽样或泊松随机抽样,可通过 R 中的帮助功能,很容易学会使用。

2.5 分层随机抽样举例

【例 6】试以文献 [5] 第 208 - 209 页“表 9 - 1”的资料作为“抽样框”,并以“sex”为分层因素,实施分层随机抽样。

解答:在 R 中,实现分层随机抽样的 strata() 函数被放置在 sampling 包中,使用前必须先下载并导入到 R 软件中。操作的步骤如下:①将安装了 R 软件的计算机联网并启动 R 软件;②选择“程序包(Packages)”并选择一个镜像(Set CRAN mirror...);③选择“安装程序包(Install package(s)...)”;④在弹出的长条形“程序包(packages)”窗口内(约有数千行,其中少部分是“程序包”绝大多数是“函数”)浏览 sampling;⑤在 R 中使用下面的语句就可实现前述的目的。

在 strata() 函数中未明确写出参数 method = "srswor",系统默认为“分层(无放回)随机抽样”。

```
> data1 <- read.table("F:/studyr/sex_age.txt", header=T)
```

```
> sub1 <- strata(data1, stratanames = "sex", size = c(4, 5), description = TRUE)
```

第一句表明,从 F 盘上文件夹为 studyr 内读取文本文件 sex_age.txt 赋值给变量(本质上是数据框) data1。

第二句表明,调用 strata() 函数,采用默认的分层随机抽样方法(即分层随机抽样);设置分层因素为 sex;分别从 F 层与 M 层抽取 4 例与 5 例受试对象或样品;需要对随机抽样出来的样品进行详细描述“description = TRUE”。

Stratum 1

Population total and number of selected units: 12 4

Stratum 2

Population total and number of selected units: 8 5

以上结果表明,从第 1 层(即 F 层) 12 个样品中随机抽取了 4 个样品;从第 2 层(即 M 层) 8 个样品中随机抽取了 5 个样品。

Number of strata 2

Total number of selected units 9

以上结果表明,总体中总共有两层,共随机抽取 9 个样品。

Warning message:

```
In strata( data1 , stratanames = " sex" , size = c( 4 , 5) , description = TRUE) :
```

```
the method is not specified; by default , the method is srswor
```

以上结果表明,系统给出了“警告信息”,在调用 strata() 函数时,关于抽样方法的参数未指定,作为默认方法,采取的是分层(无放回)随机抽样(srswor)。

```
> sub1
```

该语句表明,要求系统呈现出分层随机抽样的结果 sub1 中的与抽样有关的样品信息。

	Sex	ID	unitProb	Stratum
1	F	1	0.3333333	1
4	F	4	0.3333333	1
16	F	16	0.3333333	1
19	F	19	0.3333333	1
7	M	7	0.6250000	2
8	M	8	0.6250000	2
10	M	10	0.6250000	2
11	M	11	0.6250000	2
18	M	18	0.6250000	2

以上结果表明,第 1 列为被抽取的样品的编号、第 2 列为分层因素 sex 的水平代码、第 3 列为被抽取的样品在总体中的编号、第 4 列为各行上的样品被抽取的概率(F 层中总共 12 个样品,抽取 4 个,每一个被抽取的概率为 $4/12 = 1/3 = 0.3333333$; M 层中总共 8 个样品,抽取 5 个,每一个被抽取的概率为 $5/8 = 0.625$)、第 5 列为层的编号。

```
> getdata( data1 , sub1)
```

这一句的目的是调用 getdata() 函数,显示被抽取的样本中每个样品的全部信息(即各变量及其取值)。

	id	age	sex	ID	unitProb	Stratum
1	1	60	F	1	0.3333333	1
4	4	57	F	4	0.3333333	1
16	16	33	F	16	0.3333333	1
19	19	49	F	19	0.3333333	1
7	7	60	M	7	0.6250000	2
8	8	64	M	8	0.6250000	2
10	10	16	M	10	0.6250000	2
11	11	58	M	11	0.6250000	2
18	18	40	M	18	0.6250000	2

以上结果表明,与前面的输出结果相比较,增加了“年龄(age)”及其取值。如果原数据集(或抽样总体)中除分层因素外,还有 50 个变量及其取值,此时,都将全部显示出来。

在 strata() 函数中明确写出参数 method = "srswor",指明要进行分层(无放回)随机抽样。

```
> data1 <- read.table("F:/studyr/sex_age.txt", header = T)
> sub2 <- strata(data1, stratanames = "sex", size = c(4 5), method = "srswor", description = TRUE)
```

与前面的程序相比较,多了参数 method = "srswor"。

```
> sub2
```

	Sex	ID	unitProb	Stratum
4	F	4	0.3333333	1
6	F	6	0.3333333	1
17	F	17	0.3333333	1
19	F	19	0.3333333	1
3	M	3	0.6250000	2
7	M	7	0.6250000	2
8	M	8	0.6250000	2
11	M	11	0.6250000	2
12	M	12	0.6250000	2

以上结果表明,被抽取的样品编号发生了改变。

```
> getdata(data1, sub2)
```

	id	age	sex	ID	unitProb	Stratum
4	4	57	F	4	0.3333333	1
6	6	31	F	6	0.3333333	1
17	17	39	F	17	0.3333333	1
19	19	49	F	19	0.3333333	1
3	3	37	M	3	0.6250000	2
7	7	60	M	7	0.6250000	2
8	8	64	M	8	0.6250000	2
11	11	58	M	11	0.6250000	2
12	12	63	M	12	0.6250000	2

	Group	Age	Holders	Claims	District	ID	unitProb
1	<1L	<25	197	38	1	1	0.5
2	<1L	25~29	264	35	1	2	0.5
5	1~1.5L	<25	284	63	1	5	0.5
6	1~1.5L	25~29	536	84	1	6	0.5
.....							
18	<1L	25~29	139	19	2	18	0.5
19	<1L	30~35	151	22	2	19	0.5
17	<1L	<25	85	22	2	17	0.5
26	1.5~2L	25~29	175	46	2	26	0.5
.....							

以上结果表明,分层随机抽样的结果发生了变化。

2.6 整群随机抽样举例

2.6.1 数据集介绍

在 R 软件的 MASS 包中,有一个名为 Insurance 的数据集。其内记录了某保险公司 1973 年第三季度车险投保人的相关信息。其各观测(即行)代表投保人 5 列的列名分别为:第 1 列为 District(投保人家庭住址所在区域,代号为 1 至 4);第 2 列为 Group(所投保汽车的发动机排量,其各水平分别为:<1 升、1.0~1.5 升、>1.5~2 升、>2 升共四个等级);第 3 列为 Age[投保人的年龄,其各年龄档为:<25 岁、25~30 岁(不含 30 岁)、30~35 岁、>35 岁];第 4 列为 Holders(投保人数);第 5 列为 Claims(要求索赔的投保人数)。

2.6.2 使用 sampling 包中的 cluster() 函数进行整群随机抽样

```
> sub3 <- cluster(Insurance, clustername = "District", size = 2, method = "srswor", description = TRUE)
```

以上语句的目的是调用 cluster() 函数对 Insurance 数据集进行整群随机抽样,群的标志为“District”(即区域)、随机抽取的群数为 2、将采取简单(无放回)随机抽样(即 srswor)方法从 4 个群中随机地抽取两个群、希望对被抽取的样品信息进行描述(即 description = TRUE)。

Number of selected clusters: 2

Population total and number of selected units 64 32

以上结果表明,随机抽取了两个群,共有 64 个样品,而被抽取的样品数为 32 个。

```
> getdata(Insurance, sub3)
```

以上语句表明,调用 getdata() 函数显示从 Insurance 数据集中抽取的两个群中的全部样品。

以上是被抽取的 32 个样品,为节省篇幅,两个群内都只保留了 4 个样品,其他样品被省略了。由于共有 4 个群,因此,抽取两个群时,对于每个群而言,被抽中的概率都为 0.5。

2.7 bootstrap 随机重抽样举例

2.7.1 概述

bootstrap 随机重抽样是以原始数据为基础的模拟抽样统计推断法,其基本思想是:在原始数据范围(假定总样本含量为 n)内做有放回的再抽样,每次抽取的样本含量都是 m ,原始数据中每个观测(样品或个体)每次被抽中的概率相等,即 $1/n$,所抽得的样本被称为 bootstrap 样本。

bootstrap 样本有何用途呢?当人们无法驾驭总体但可以反复获得 bootstrap 样本时,由它们所呈现的规律就非常接近无法驾驭的那个总体。下面用一个可以驾驭的总体为例,先呈现出总体中一个定量变量的取值的频数分布情况,再采取有放回的随机抽样,获得一个样本含量较大(如 1000 或 10000)的 bootstrap 样本,再呈现该 bootstrap 样本的频数分布。比较这两个频数分布的形状是否接近。

2.7.2 用于抽取 bootstrap 样本的数据集

在 R 中,有一个内置的 faithful 数据集(Old Faithful Geyser Data),它记录了美国怀俄明州黄石国家公园内的古老而又忠实的喷泉的喷射持续时间 eruptions(min) 和等待时间 waiting(min) (即相邻两次喷射之间的时间间隔)。

```
> faithful
```

此语句用于显示 faithful 数据集中的全部数据 ($n = 272$)。

	eruptions	waiting
1	3.600	79
2	1.800	54
3	3.333	74
.....		
270	4.417	90
271	1.817	46
272	4.467	74
.....		

以上显示出 faithful 数据集中的开始和结尾的各 3 个观测结果。

2.7.3 显示总体中 eruptions 变量的频数分布

```
> attach( faithful)
```

```
> hist( eruptions ,breaks = 25)
```

第一句的目的是绑定数据;第二句的目的是绘制 eruptions 变量的频数直方图,因篇幅所限,图形从略。

2.7.4 从总体中进行 2000 次有放回的随机抽样,获得样本含量 $n = 2000$ 的 bootstrap 样本

```
> boot.sample <- sample( eruptions 2000 rep = T)
```

以上语句的目的就是为了实现从总体中进行 2000 次有放回的随机抽样,获得样本含量 $n = 2000$ 的 bootstrap 样本。

2.7.5 将总体与 bootstrap 样本中的 eruptions 变量的频数分布直方图平行地绘制出来

```
> par( mfrow = c( 1 2) )
```

```
> hist( eruptions ,breaks = 25)
```

```
> hist( boot.sample ,breaks = 25)
```

```
> par( mfrow = c( 1 1) )
```

以上四个语句的目的分别为:

第一句:设置作图窗口为一行两列(就是希望把两幅图左右放置)。

第二句:在左边放置总体中的 eruptions 变量的频数分布直方图。

第三句:在右边放置 bootstrap 样本中的 eruptions 变量的频数分布直方图。

第四句:设置作图窗口为一行一列(就是把前面的一行上绘制两幅图的设置改回默认状态,即一行上只绘制一幅图)。

由图形绘制的结果(因篇幅所限,此处从略)可以清楚地看出:bootstrap 样本呈现的图形几乎与总体一模一样。若总体真是不可驾驭的,那就可以通过 bootstrap 样本呈现的形状去推断它。

参考文献

- [1] 黄文,王正林. 数据挖掘: R 语言实战[M]. 北京: 电子工业出版社, 2015: 34 - 39.
- [2] 李诗羽,张飞,王正林. 数据分析: R 语言实战[M]. 北京: 电子工业出版社, 2015: 88 - 156.
- [3] 方匡南,朱建平,姜叶飞. R 数据分析: 方法与案例详解[M]. 北京: 电子工业出版社, 2015: 54 - 168.
- [4] Joseph Adler. R 语言核心技术手册[M]. 2 版. 刘思喆,李舰,陈钢,等译. 北京: 电子工业出版社, 2015: 417 - 421.
- [5] 胡良平. 科研设计与统计分析[M]. 北京: 军事医学科学出版社, 2012: 129 - 274.

(收稿日期: 2016 - 12 - 03)

(本文编辑: 陈 霞)